

NAS

COUPLING IN

FLEXIBLE ROBOTIC MANIPULATORS

by

Dr. Dalbert Tolar
Cockrell Center
in Engineering

Philip Leo Graves
Research Assistant

Mechanical Engineering
UNIVERSITY OF TEXAS AT AUSTIN
Austin, Texas 78712
(512) 471-3012

August, 1988

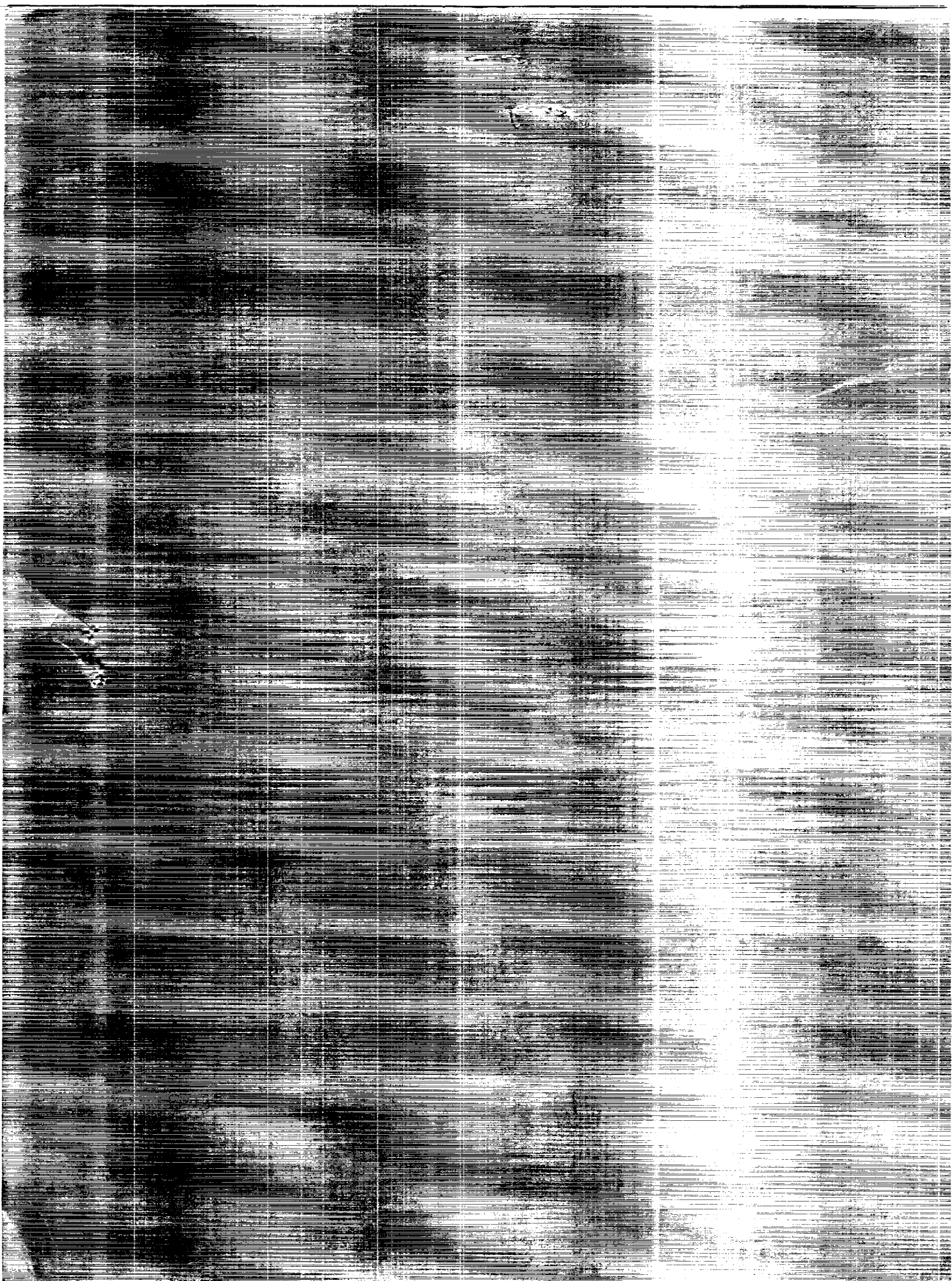
(NASA-CR-187679) THE EFFECT OF INERTIAL
COUPLING IN THE DYNAMICS AND CONTROL OF
FLEXIBLE ROBOTIC MANIPULATORS Final Report
(Texas Univ.) 165 p

CSCL 13T

N91-13721

Unclass

63/37 0319626



**THE EFFECT OF INERTIAL COUPLING IN THE DYNAMICS AND
CONTROL OF FLEXIBLE ROBOTIC MANIPULATORS**

by

**Dr. Delbert Tesar
Carol Cockrell Curran
Chair in Engineering**

**Philip Lee Graves
Graduate Research Assistant**

**Department of Mechanical Engineering
UNIVERSITY OF TEXAS AT AUSTIN
Austin, Texas 78712
(512) 471-3039**

August, 1988

ABSTRACT

Flexibility occurs in all mechanisms, to some degree. The dynamics of most flexible robotic manipulators can be modeled by a system of coupled ordinary differential equations. This study presents a general model of the dynamics of flexible robotic manipulators, including the gross motion of the links, the vibrations of the links and joints, and the dynamic coupling between the gross motions and vibrations.

The vibrations in the links may be modeled using lumped parameters, truncated modal summation, a component mode synthesis method, or a "mixture" of these methods. The local link inertia matrix is derived to obtain the coupling terms between the gross motion of the link and the vibrations of the link. Coupling between the motions of the links results from the kinematic model, which utilizes the method of kinematic influence.

The model is used to simulate the dynamics of a flexible, space-based robotic manipulator which is attached to a spacecraft, and is free to move with respect to the inertial reference frame. This model may be used to study the dynamic response of the manipulator to the motions of its joints, or to externally applied disturbances.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	v
ABSTRACT.....	vi
TABLE OF CONTENTS.....	viii
LIST OF TABLES.....	xi
LIST OF FIGURES.....	xii
1.0 Introduction.....	1
1.1 Literature Review.....	2
2.0 The Reference Model.....	5
2.1 Geometry of a Flexible Robotic System.....	6
2.1.1 Angular Displacements and Deflections.....	8
2.1.2 Translational Displacements and Deflections.....	15
2.1.3 Position of the Center of Mass.....	17
2.2 Kinematics of a Flexible Robotic System.....	18
2.2.1 General Kinematic Foundation.....	20
2.2.2 First Order Kinematic Influence.....	21
2.2.2.1 Translations.....	22
2.2.2.2 Rotations.....	24

2.2.2.3 Translations and Rotations	
Due to the Motion of the	
Global Frame.....	25
2.2.2.4 Matrix Notation and the	
Jacobian Matrix....	25
2.2.3 Second Order Kinematic Influence.....	26
2.2.3.1 Second Order Translations.....	27
2.2.3.2 Second Order Rotations.....	28
2.2.3.3 Second Order Kinematic	
Effects Due to Gross Motion.....	29
2.2.3.4 Matrix Notation and the	
Hessian Matrix.....	30
2.2.4 Local Link Kinematics.....	30
2.2.4.1 Local Link Velocities	
and Accelerations.....	31
2.2.5 Kinematics of the Center	
of Mass of the System.....	32
2.2.6 Kinematic Models and	
Momentum Conservation.....	33
2.3 The Dynamics of the System.....	35
2.3.1 Vibration Model Concepts.....	36
2.3.2 Kinetic Energy and the	
System Inertia Matrix.....	39

2.3.3 Potential Energy and the	
Stiffness Matrix.....	44
2.3.4 The Equations of Motion and	
the Power Modeling Matrix.....	46
2.3.5 Applied Forces and Torques.....	54
3.0 Dynamic Simulations.....	56
3.1 Example: A Large Space-Based Robotic System.....	57
4.0 Conclusions.....	96
APPENDIX A: Derivation of the Local Inertia Matrix.....	98
APPENDIX B: Derivation of the Stiffness Matrix.....	120
APPENDIX C: VSim: A Simulation Package.....	122
REFERENCES.....	154

LIST OF TABLES

3.1-1 Geometric Data for Example Manipulator.....	58
3.1.2 Mass and Inertia for Example Manipulator.....	59

LIST OF FIGURES

2.1-1	The Reference Frames.....	7
2.1-2	Angular Link Parameters.....	9
2.1-3	Translational Link Parameters.....	10
2.1-4	Position Vectors.....	10
2.1-5	Local Link Deflections.....	11

(The Following Figures are the Results of Simulations)

3.1-1	Link 1, (a) Mode 1, (b) Mode 2.....	71
3.1-2	Link 1, (a) Mode 3, (b) Mode 4.....	72
3.1-3	Link 2, (a) Mode 1, (b) Mode 2.....	73
3.1-4	Link 2, (a) Mode 3, (b) Mode 4.....	74
3.1-5	Link 1, (a) Mode 1, (b) Mode 2.....	75
3.1-6	Link 1, (a) Mode 3, (b) Mode 4.....	76
3.1-7	Link 2, (a) Mode 1, (b) Mode 2.....	77
3.1-8	Link 2, (a) Mode 3, (b) Mode 4.....	78
3.1-9	Position of Resolution Position.....	79
3.1-10	Link 1, (a) Mode 1, (b) Mode 2.....	80
3.1-11	Link 1, (a) Mode 3, (b) Mode 4.....	81
3.1-12	Link 2, (a) Mode 1, (b) Mode 2.....	82
3.1-13	Link 2, (a) Mode 3, (b) Mode 4.....	83
3.1-14	Link 1, (a) Mode 1, (b) Mode 2.....	84
3.1-15	Link 1, (a) Mode 3, (b) Mode 4.....	85

3.1-16	Link 2, (a) Mode 1, (b) Mode 2.....	86
3.1-17	Link 2, (a) Mode 3, (b) Mode 4.....	87
3.1-18	Link 1, (a) Mode 1, (b) Mode 2.....	88
3.1-19	Link 1, (a) Mode 3, (b) Mode 4.....	89
3.1-20	Link 2, (a) Mode 1, (b) Mode 2.....	90
3.1-21	Link 2, (a) Mode 3, (b) Mode 4.....	91
3.1-22	Link 1, (a) Mode 1, (b) Mode 2.....	92
3.1-23	Link 1, (a) Mode 3, (b) Mode 4.....	93
3.1-24	Link 2, (a) Mode 1, (b) Mode 2.....	94
3.1-25	Link 2, (a) Mode 3, (b) Mode 4.....	95

1.0 Introduction

All real mechanical devices exhibit some flexibility. If the flexibility is negligible, the system dynamics can be modeled by a set of second order differential equations, developed from Newton's laws of motion. But for other compliant systems, this model is not sufficient. The structural dynamics, and the coupling between the structural and gross motions of the system must be included to obtain a model of the system dynamics.

Models of vibrations in continuous systems usually assume that the structure is homogeneous, isotropic, and obeys Hooke's law within the elastic limit [23]. The dynamics of the system is represented by partial differential equations, such as the Euler equation for beams. This form of equation is not suitable for real-time computation of the dynamic model. Therefore mode summation methods, component mode synthesis methods, and lumped parameter methods are used. These methods use second order ordinary differential equations to model vibrations, and are sufficient to describe the vibrations in almost all flexible manipulators.

The first goal of this study is to formulate a total model of the dynamics of a flexible manipulator system, which is valid for a wide variety of flexible robotic systems. The second goal is to analyze the dynamic coupling which occurs between the

gross motion and vibrations in the system. The final goal is to present specific applications of the model which utilize the knowledge of dynamic coupling to study the system dynamics.

1.1 Literature Review

Research in the dynamics of robotics and other spatial mechanisms is a union of geometry, the kinematics of mechanisms, and classical mechanics, each drawing on the knowledge of the other two. Robotics, as a broader subject, involves principles from almost every field of engineering and science. Researchers in the field have differing backgrounds, and therefore many methods and perspectives have been published about the various subjects in the field. This review will cover the major papers relevant to the dynamics of flexible manipulators.

Benedict and Tesar [1-3] introduced the concept of kinematic influence coefficients, which were applied to rigid and quasi-rigid planar mechanisms. Sanders and Tesar [22] showed via experimentation that the quasi-static assumption is valid for mechanisms which operate well below the first natural frequency of the mechanism. The concept of kinematic influence was generalized for spatial mechanisms with rigid links by Thomas and Tesar [4]. Freeman and Tesar [5] showed that this

method is extremely powerful and may be used to model the dynamics of both serial and parallel robotic manipulators. Fresonki, Henandez, and Tesar [6][25] used the concept of kinematic influence to obtain a description of the spatial deflections in robotic mechanisms. Kinematic influence was used by Behi and Tesar [7] to model vibrations in a multi-degree of freedom system due to flexibilities in the drive mechanisms. Wander and Tesar [8] proved that a totally general dynamic model of a robotic manipulator may be computed in real-time using the method of kinematic influence coefficients.

Denavit and Hartenberg [9] are well know for a kinematic notation used to describe spatial, multi-degree of freedom devices. Most kinematics notational schemes seen in the literature vary little from this scheme. A recursive algorithm for computing the dynamics of rigid manipulators was presented by Hollerbach [10]. Book [11-15], Maizza-Neto [12,16], and Whitney, [12,15] and their associates were some of the first to study the dynamics of flexible manipulators. Book [14] formulated a recursive Lagrangian algorithm with a truncated mode summation representation of vibrating links, as did King, Gourishankar, and Rink [22]. In [16], Hughes developed a model of the space shuttle manipulator arm. Hamilton's principle was used by Low [17] to develop the explicit equations of motion for

a manipulator with flexible links. Finite element methods have been used by Sunada and Dubowsky [18], and by Naganathan and Soni [19] to analyze manipulators with elastic links. Huang and Lee [20] extended the Newton-Euler formulation of dynamics to model non-rigid manipulators.

2.0 The Reference Model

The dynamic model of a flexible manipulator presented here is a general method which may be used to model the majority of flexible manipulators. Drive mechanism flexibilities may be modeled using lumped parameters. Structural flexibilities may be modeled using lumped parameters, assumed modes (a truncated mode summation technique), a component mode synthesis technique, or a combination of these methods. The result is a set of ordinary coupled differential equations, which model the dynamics of the system, including the gross motions, the vibrations, and the coupling between the gross motions and vibrations.

All models of vibrations are approximations. Due to the generality of the model, and the variety of methods which may be used to represent the vibrations, it is possible to create a model which is a much closer approximation to the real system, than it would if only one method was used to represent the vibrations. To do so, the actual deflections of the system must be observed, or predicted to assure that the vibrations model is correct. The reader unfamiliar with models of vibrations in continuous systems is urged to refer to Thomson [23].

2.1 Geometry of a Flexible Robotic System

In a spatial manipulator consisting of $n+1$ distinct links, a local body-fixed reference frame is assigned to each link. A requirement of Newtonian dynamics is that all motion must be measured relative to an inertial reference frame. If the base link of the manipulator is fixed relative to the inertial reference frame, the reference system of the base link becomes the inertial frame. Otherwise, a frame which is not attached to the manipulator will be the inertial reference frame. Thus, for the constrained case there will be $n+1$ reference frames, and for the unconstrained case there will be $n+2$. One of the reference frames is chosen to be the global reference frame, and is denoted as frame h , as shown in Figure 2.1-1. The global frame serves as a common frame to which all vector quantities will be referenced. Notice that the global frame is not required to be the same as the inertial frame. A preceding superscript enclosed in parentheses is used to denote the local frame to which a vector is referenced. (ie. ${}^{(1)}\underline{R}$ is referenced to frame 1.) If the superscript is not shown, the vector is assumed to be referenced to frame h .

The geometry of the manipulator is described by the instantaneous orientations and positions of the reference frames, shown in Figure 2.1-1. Frame 1 is attached to link 1. The unit vectors $\hat{\underline{S}}_1^x$, $\hat{\underline{S}}_1^y$, and $\hat{\underline{S}}_1^z$ are the direction cosines of the

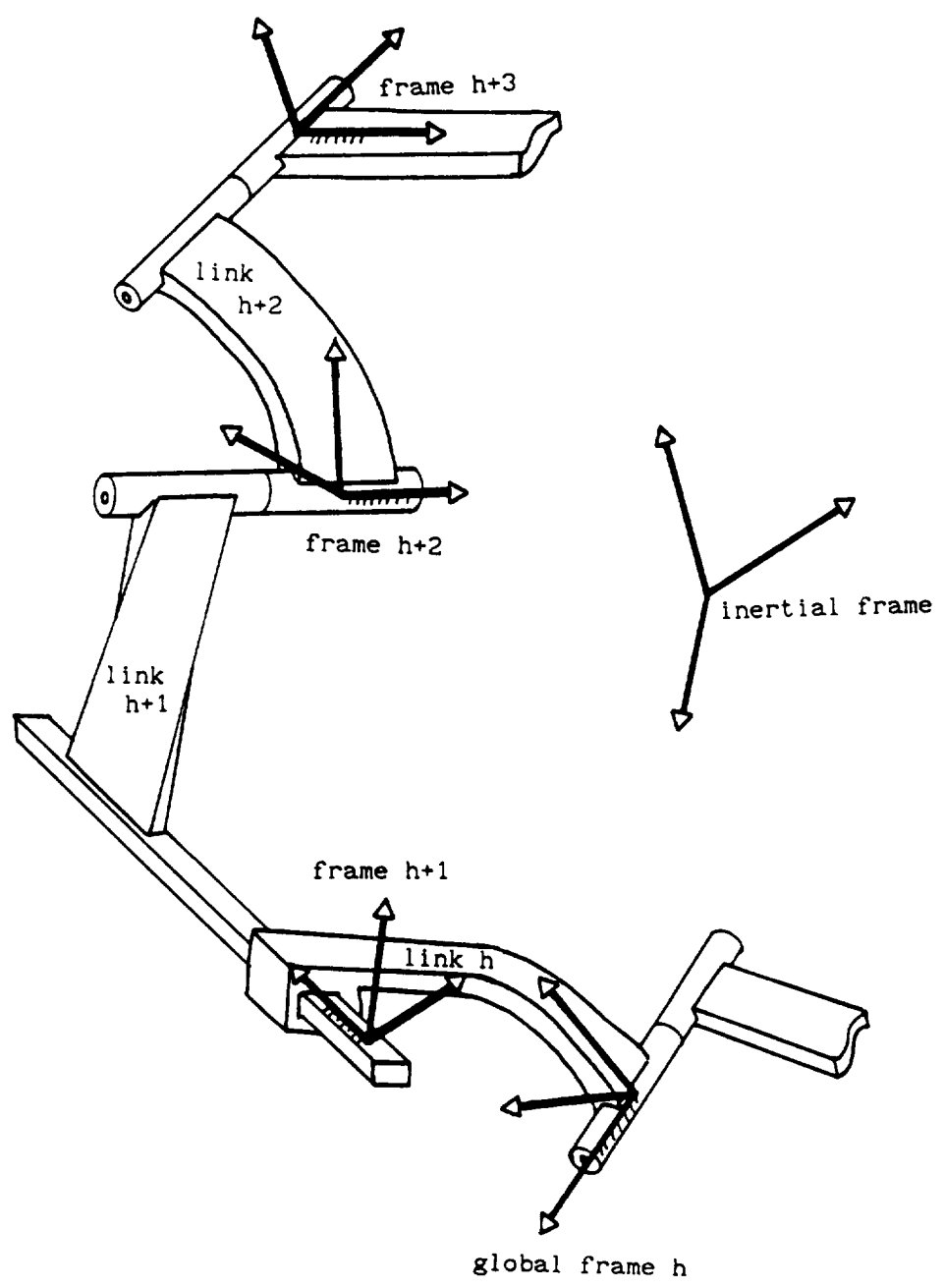


Figure 2.1-1 The Reference Frames

x, y, and z axes of frame 1, defining the orientation. The unit vector $\hat{\underline{S}}_1^z$ coincides with the line of action of joint 1. Joint 1 may be a revolute or prismatic juncture between links i-1 and i. $\hat{\underline{S}}_1^x$ is a common perpendicular to the successive joint axes $\hat{\underline{S}}_1^z$ and $\hat{\underline{S}}_{i+1}^z$, when no deflection occurs in link i.

The undeflected position of a flexible link is considered to be the reference position. When link i is in its reference position, the direction cosines of local frame i+1 are denoted as $\hat{\underline{S}}_1^{x'}$, $\hat{\underline{S}}_1^{y'}$, and $\hat{\underline{S}}_1^{z'}$. The vector \underline{x}_1 defines the position of a point on link i, relative to the origin of frame 1. Deflections of the point are described by the vector functionals $\underline{d}(\underline{x}_1)$, and $\theta(\underline{x}_1)$, which are the linear and angular deflections relative to the reference position.

2.1.1 Angular Displacements and Deflections

The relative orientation of successive local coordinate frames is defined by a set of ordered rotations, shown in Figure 2.1-2. The angular deflections of the link are assumed to be small, and thus add vectorially. This first order approximation of the rotation is necessary in three dimensional systems where there is no dominant angular deflection in one direction, and will introduce errors into the geometry calculations smaller than 5% if the magnitude of the angular deflections is less than 3 degrees. The angular deflection of a point on link i from its undeflected orientation is represented

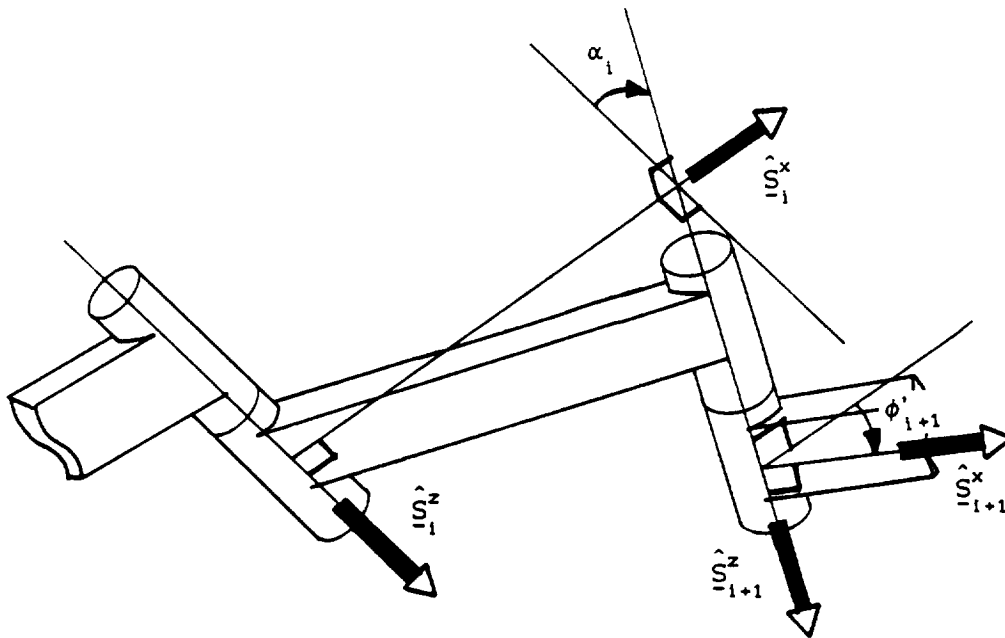


Figure 2.1-2 Angular Link Parameters

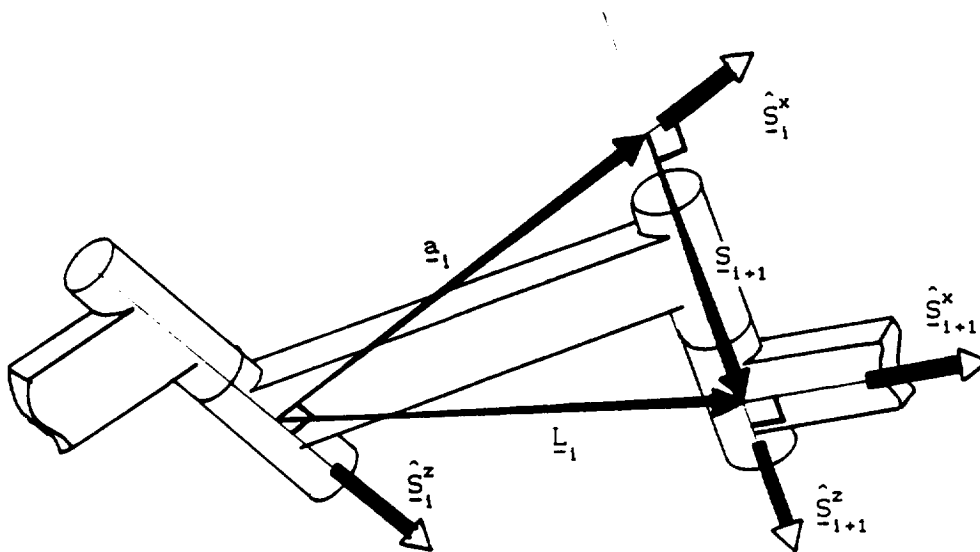


Figure 2.1-3 Translational Link Parameters

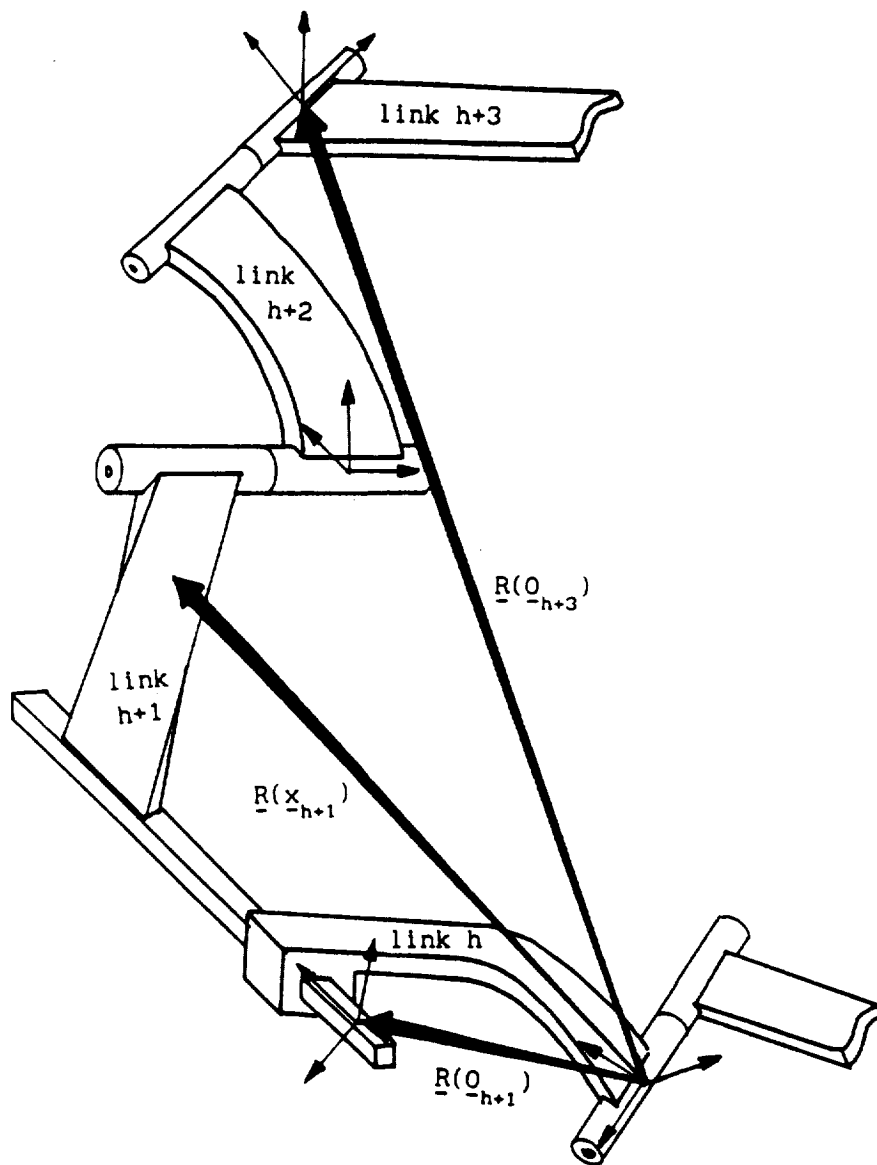


Figure 2.1-4 Position Vectors

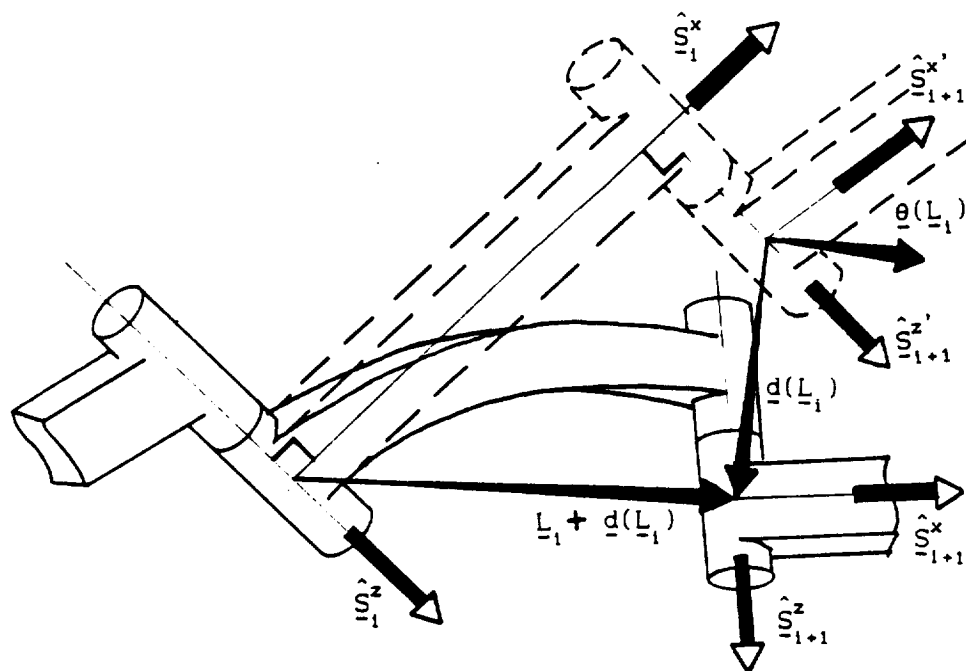


Figure 2.1-5 Local Link Deflections

by the vector functional $\underline{\theta}(\underline{x}_i)$ which describes the relative translational deflection of the distal end of link i from its reference position. Let $\psi_j(\underline{x}_i)$ be the shape function describing the rotational deflection of mode j of link i , and q_{ij} be the corresponding generalized coordinate which is the magnitude of the mode. The functional relationship between the total angular deflection of a small element on a link with m_i modes is:

$$^{(1)}\underline{\theta}(\underline{x}_i) = \sum_{j=1}^{m_i} \left\{ q_{ij} \psi_j(\underline{x}_i) \right\} \quad (2.1-1)$$

or, in matrix form:

$$^{(1)}\underline{\theta}(\underline{x}_i) = \left[\psi(\underline{x}_i) \right] \underline{q}_i \quad (2.1-2)$$

When the elements of the rotation are organized into a skew-symmetric form, the small rotation matrix results.

$$[T\theta(\underline{x}_i)] = \begin{bmatrix} 1 & -\theta_z(\underline{x}_i) & \theta_y(\underline{x}_i) \\ \theta_z(\underline{x}_i) & 1 & -\theta_x(\underline{x}_i) \\ -\theta_y(\underline{x}_i) & \theta_x(\underline{x}_i) & 1 \end{bmatrix} \quad (2.1-3)$$

These rotations are measured in the local i th coordinate system.

Notice that the determinant of this small rotation matrix is

$$\det[T\theta(\underline{x}_i)] = 1 + |\underline{\theta}(\underline{x}_i)| \approx 1, \quad (2.1-4)$$

because $|\underline{\theta}(\underline{x}_i)|$ is small, as assumed previously. The inverse of

the small rotation matrix can then be approximated as the transpose:

$$[T\theta(\underline{x}_i)]^{-1} \approx [T\theta(\underline{x}_i)]^T \quad (2.1-5)$$

Next, the angle α_i is defined about the resulting x-axis. The new z-axis, formed by these rotations, is parallel to the \hat{S}_{i+1}^z axis, which is the line of action of joint i+1. The final rotation is ϕ'_{i+1} about \hat{S}_{i+1}^z , where ϕ'_{i+1} is the sum of the gross displacement of joint i+1, ϕ_{i+1} , and the deflection of the joint, $\delta\phi_{i+1}$. The resultant transformation from coordinate system i to coordinate system i+1 is:

$$[{}^{i+1}T_i] = [T\theta(\underline{L}_i)] \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\alpha_i) & -s(\alpha_i) \\ 0 & s(\alpha_i) & c(\alpha_i) \end{bmatrix} \begin{bmatrix} c(\phi'_{i+1}) & -s(\phi'_{i+1}) & 0 \\ s(\phi'_{i+1}) & c(\phi'_{i+1}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1-6)$$

where $s()$ \equiv sine() and $c()$ \equiv cosine().

The matrix hT_i transforms a vector from the local coordinate system of link i to the coordinate system of the reference link, h . The columns of hT_i are composed of the direction cosines of the unit vectors \hat{S}_i^x , \hat{S}_i^y , and \hat{S}_i^z , and are obtained via the formula:

$${}^hT_i = \begin{bmatrix} {}^{(h)}\hat{S}_i^x & {}^{(h)}\hat{S}_i^y & {}^{(h)}\hat{S}_i^z \end{bmatrix}$$

$$= \prod_{j=h}^{i-1} {}^jT_{j+1} \quad \text{for } i > h$$

$$= [I] \text{ (identity matrix) for } i = h$$

$$= \prod_{j=i}^{h-1} {}^jT_{j+1} \quad \text{for } i < h \quad (2.1-7)$$

2.1.2 Translational Displacements and Deflections

The reference position of the distal end of the link is \underline{L}_i , and is the sum of the vector \underline{a}_i directed along $\hat{\underline{S}}_i^x$, and the vector \underline{S}_{i+1} which is directed along $\hat{\underline{S}}_{i+1}^z$, as shown in Figure 2.1-3. The vector functional $\underline{d}(\underline{L}_i)$ describes the relative translational deflection of the distal end of link i from its reference position. Let $\delta_j(\underline{x}_i)$ be the shape function, or mode shape describing the translational deflection of mode j of link i , and $q_{i,j}$ be the corresponding generalized coordinate which is the magnitude of the mode. The functional relationship between the generalized coordinates and the translational deflection of a link with m_i modes is:

$${}^{(1)}\underline{d}(\underline{x}_i) = \sum_{j=1}^{m_i} \left\{ q_{i,j} \delta_j(\underline{x}_i) \right\} \quad (2.1-8)$$

or, in matrix form:

$${}^{(1)}\underline{d}(\underline{x}_i) = \left[\delta(\underline{x}_i) \right] \underline{q}_i \quad (2.1-9)$$

The dimension of the modal matrix is $3 \times m_i$, and the columns are the translational parts of the mode shape functions.

The vector equations describing the position of a point on link i relative to the origin of link h (for a serial chain) are:

FOR $i > h$:

$$\begin{aligned} \underline{R}(\underline{x}_i) &= \sum_{j=h}^{i-1} [{}^hT_j] \left\{ {}^{(j)}\underline{L}_j + {}^{(j)}\underline{d}(\underline{L}_j) \right\} + [{}^hT_i] \left\{ {}^{(1)}\underline{x}_i + {}^{(1)}\underline{d}(\underline{x}_i) \right\} \\ &= \sum_{j=h}^{i-1} [{}^hT_j] \left\{ {}^{(j)}\underline{S}_j + {}^{(j)}\underline{a}_j + \left[\delta(\underline{x}_j) \right] \underline{q}_j \right\} \\ &\quad + [{}^hT_i] \left\{ {}^{(1)}\underline{x}_i + \left[\delta(\underline{x}_i) \right] \underline{q}_i \right\} \end{aligned}$$

FOR $i = h$:

$$\underline{R}(\underline{x}_i) = \underline{x}_i + \left[\delta(\underline{x}_i) \right] \underline{q}_i \quad (2.1-10)$$

FOR $i < h$:

$$\begin{aligned} \underline{R}(\underline{x}_i) &= \sum_{j=h-1}^i [{}^hT_j] \left\{ {}^{(j)}\underline{S}_j + {}^{(j)}\underline{a}_j + \left[\delta(\underline{x}_j) \right] \underline{q}_j \right\} \\ &\quad + [{}^hT_i] \left\{ {}^{(1)}\underline{x}_i + \left[\delta(\underline{x}_i) \right] \underline{q}_i \right\} \end{aligned}$$

These vectors are shown in Figure 2.1-4.

2.1.3 Position of the Center of Mass

Also of geometric interest is the equation of the center of mass of the system. Let \underline{cm}_1 be the vector from the origin of link 1 to the center of mass of link 1, in the reference position. Let μ_1 be the ratio of the mass of link 1 to the total mass of the system. For a rigid system, the position of the center of mass is:

$$\underline{R}_{cm} = \sum_{i=0}^n \mu_i \underline{cm}_i + \sum_{j=h+1}^n \left\{ \mu_j \left[\sum_{k=h}^{j-1} \underline{L}_k \right] \right\} - \sum_{j=0}^{h-1} \left\{ \mu_j \left[\sum_{k=j}^{h-1} \underline{L}_k \right] \right\} \quad (2.1-11)$$

Although the derivation of this equation is not given here, the reader may verify it by noting that:

$$\underline{R}_{cm} \sum_{i=0}^n m_i = \sum_{i=0}^n \left\{ m_i \underline{R}(\underline{cm}_i) \right\}. \quad (2.1-12)$$

2.2 Kinematics of a Flexible Robotic System

Kinematics is a study of how motions interact based on the geometric properties of a mechanism, ignoring the forces which may occur. The complete kinematic description of a serial chain of compliant bodies includes the motion of the actuated joints, the motion due to vibrations, and the motion of the whole system relative to an inertial reference frame.

Thomas, Tesar, Freeman, [4-5] and their associates have developed a general kinematic representation to describe actuated motion in rigid-body mechanisms, for both serial and parallel link topologies. Fresonki, Behl, and Tesar [6-7] have used the concepts developed for rigid body kinematics in a model of the kinematics of a compliant system, to predict the deflections and frequency response of the mechanism.

In this model, the concept of kinematic influence is extended to ascertain the total kinematic effect of motions due to compliance of both joints and links. The gross motion of the whole system relative to an inertial reference frame is also examined. The final result is a general kinematic model based on the kinematic influence of all possible motions in a compliant system.

The relative motion between bodies is defined by the relative motions of the local reference frames. Each motion is defined by a line of action, \hat{S} , and a magnitude, $\dot{\phi}$. The line of

action of joint 1 between links i-1 and i is defined by $\hat{\underline{S}}_1^z$, and the magnitude is defined by $\dot{\phi}_1$. The motions due to the vibrations are defined by the three orthogonal unit vectors $\hat{\underline{S}}_1^x$, $\hat{\underline{S}}_1^y$, and $\hat{\underline{S}}_1^z$, and by the mode shape and magnitude of each mode of vibration.

The kinematic foundation used to model the spatial deflections of the system, and the motion of the system relative to the inertial reference frame, is similar to the kinematic foundation used to model the motion due to the actuated joints. The original development of this kinematic method was performed by Tesar, and his associates [1-7] and was used to model mechanisms with no flexibility, which were fixed relative to the inertial reference frame. The degrees of freedom in these systems were always associated with an actuated joint, and thus the term input became synonymous with the term degree of freedom. Behi, Fresonki, and Tesar extended this model to include deflections of the system. In order to emphasize the fact that the spatial deflections can be modeled in the same manner as the actuated joints, the spatial deflections have been called pseudo-inputs. In this model, the spatial deflections of the link $\underline{d}(\underline{x}_i)$ and $\underline{\theta}(\underline{x}_i)$, and the joint deflection are pseudo-inputs. The pseudo-inputs are related to the generalized coordinates of vibrations via the modal matrix.

2.2.1 General Kinematic Foundation

Many popular kinematic formulations are based on a recursive algorithm which transforms velocities in a serial fashion from local coordinate system i to $i+1$, adding the contribution of local motions in an iterative procedure. Kinematics may be approached in a more efficient way by utilizing the geometric information of the system, allowing the velocity contribution of each motion to be computed independently. [8]

The velocity of a point P on link i is the vector sum of the velocity contributions of all motions between the inertial reference frame and the point P . The velocity contribution of a single motion is the magnitude of that motion multiplied by a kinematic influence coefficient. The kinematic influence coefficient is a vector derivative, and is readily obtained from the geometric description of the mechanism. By organizing the kinematic influence coefficients into a matrix, the Jacobian is formed.

Each type of motion which may occur, may be considered to be a relative velocity, and may be modeled by this method. The gross motion of the system, the joint motions, and the spatial deflections each are multiplied by an appropriate kinematic influence coefficient. The sum of all these relative motions gives the "absolute" velocity.

2.2.2 First Order Kinematic Influence

The first order kinematic influence coefficient is defined as the rate of change of the position or orientation of a point with respect to a position state. A position state may be the gross position of the mechanism, the position of an actuated joint, or the deflection of the link or joint. The "absolute" translational velocity of a point at position \underline{x}_k on link k is the sum of the kinematic influence coefficients multiplied by the corresponding velocity states:

$$\dot{\underline{R}}(\underline{x}_k) = \sum_{i=1}^{\text{dof}} \left[\partial \underline{R}(\underline{x}_k) / \partial \varphi_i \dot{\varphi}_i \right] = \sum_{i=1}^{\text{dof}} \left[\underline{g}_i^P(\underline{x}_k) \dot{\varphi}_i \right], \quad (2.2-1)$$

where dof = the total number of degrees of freedom in the system, and $\underline{g}_i^P(\underline{x}_k)$ is the translational kinematic influence coefficient. The "absolute" rotational velocity is

$$\underline{\omega}(\underline{x}_k) = \sum_{i=1}^{\text{dof}} \left[\hat{\underline{S}}_i \dot{\varphi}_i \right] = \sum_{i=1}^{\text{dof}} \left[\underline{g}_i^R(\underline{x}_k) \dot{\varphi}_i \right], \quad (2.2-2)$$

where $\underline{g}_i^R(\underline{x}_k)$ is the rotational kinematic influence coefficient.

2.2.2.1 Translations

The kinematic influence of a translational motion on the translation of a point P (in a serial mechanism) is the $\hat{\underline{S}}_j$ vector describing the line of action of the motion. For a translational motion designated as $\dot{\phi}_j$, and a point P at local reference position \underline{x}_k , the kinematic influence coefficient is defined by the following:

IF the translational motion j causes point P to translate relative to the global frame h, AND $k \geq h$:

$${}^{(h)}\underline{g}_j^P(\underline{x}_k) = {}^{(h)}\hat{\underline{S}}_j \quad (2.2-3a)$$

IF the translational motion j causes point P to translate relative to the global frame h, AND $k < h$:

$${}^{(h)}\underline{g}_j^P(\underline{x}_k) = -{}^{(h)}\hat{\underline{S}}_j. \quad (2.2-3b)$$

For all other cases,

$${}^{(h)}\underline{g}_j^P(\underline{x}_k) = \underline{0} \quad (2.2-3c)$$

The kinematic influence of a rotational motion on the translation of a point (in a serial mechanism) is the cross product of the line of action of the rotation, $\hat{\underline{S}}$, and a vector from the line of action of the motion to the point P. For a rotational motion at local reference position \underline{x}_j , and a point P at local reference position \underline{x}_k , the kinematic influence is defined by the following:

IF the rotational motion j causes point P to translate relative to the global frame h, AND $k \geq h$:

$${}^{(h)}\underline{g}_j^P(\underline{x}_k) = {}^{(h)}\hat{\underline{S}}_j \times \left\{ {}^{(h)}\underline{R}(\underline{x}_k) - {}^{(h)}\underline{R}(\underline{x}_j) \right\}. \quad (2.2-4a)$$

IF the rotational motion j causes point P to translate relative to the global frame h, AND $k < h$:

$${}^{(h)}\underline{g}_j^P(\underline{x}_k) = {}^{(h)}\hat{\underline{S}}_j \times \left\{ {}^{(h)}\underline{R}(\underline{x}_j) - {}^{(h)}\underline{R}(\underline{x}_k) \right\}. \quad (2.2-4b)$$

For all other cases,

$${}^{(h)}\underline{g}_j^P(\underline{x}_k) = 0 \quad (2.2-4c)$$

2.2.2.2 Rotations

The kinematic influence of a rotational motion on the rotation of a differential mass element surrounding point P (in a serial mechanism) is the vector $\hat{\underline{S}}_j$ describing the line of action of the motion. For a rotational motion at local reference position \underline{x}_j , and an element P at local reference position \underline{x}_k , the kinematic influence coefficient is defined by the following:

IF the rotational motion j causes the differential element surrounding point P to rotate relative to the global reference frame h, AND $k \geq h$:

$${}^{(h)}\underline{g}_j^R(\underline{x}_k) = {}^{(h)}\hat{\underline{S}}_j \quad (2.2-5a)$$

IF the rotational motion j causes the differential element surrounding point P to rotate relative to the global reference frame h, AND $k < h$:

$${}^{(h)}\underline{g}_j^R(\underline{x}_k) = -{}^{(h)}\hat{\underline{S}}_j \quad (2.2-5b)$$

For all other cases,

$${}^{(h)}\underline{g}_j^R(\underline{x}_k) = 0 \quad (2.2-5c)$$

2.2.2.3 Translations and Rotations Due to Motion Of The Global Frame

Systems with bases which move relative to the inertial reference frame undergo gross motion which must be included in the dynamics of the system, therefore they will be included in the kinematic model of the system. These motions are easily described as three orthogonal translations and three orthogonal rotations at the global frame.

2.2.2.4 Matrix Notation and the Jacobian Matrix

The kinematic influence coefficients for a point P can be organized into matrix form (a Jacobian):

$${}^{(h)}\dot{\underline{R}}(\underline{x}_1) = \left[{}^{(h)}\underline{G}^P(\underline{x}_1) \right] \dot{\underline{\varphi}} \quad (2.2-6)$$

and

$${}^{(h)}\dot{\underline{\omega}}_1(\underline{x}_1) = \left[{}^{(h)}\underline{G}^R(\underline{x}_1) \right] \dot{\underline{\varphi}} \quad (2.2-7)$$

Or in a more compact notation,

$$\left\{ \begin{array}{l} {}^{(h)}\dot{\underline{R}}_1(\underline{x}_1) \\ {}^{(h)}\dot{\underline{\omega}}_1(\underline{x}_1) \end{array} \right\} = \left[{}^{(h)}\underline{G}(\underline{x}_1) \right] \dot{\underline{\varphi}} \quad (2.2-8)$$

2.2.3 Second Order Kinematic Influence

The second order kinematic relationships for a serial chain are defined by the second derivative of the position:

$$\begin{aligned}
 \ddot{\underline{R}} &= d/dt \left\{ \sum_{i=1}^{\text{dof}} \left[\underline{g}_i^P \dot{\varphi}_i \right] \right\} \\
 &= \sum_{i=1}^{\text{dof}} \left[\underline{g}_i^P \ddot{\varphi}_i \right] + \sum_{i=1}^{\text{dof}} \sum_{j=1}^{\text{dof}} \left\{ \partial/\partial \varphi_j \left[\underline{g}_i^P \right] \dot{\varphi}_i \dot{\varphi}_j \right\} \\
 &= \sum_{i=1}^{\text{dof}} \left[\underline{g}_i^P \ddot{\varphi}_i \right] + \sum_{i=1}^{\text{dof}} \sum_{j=1}^{\text{dof}} \left\{ \underline{h}_{ij}^P \dot{\varphi}_i \dot{\varphi}_j \right\} \quad (2.2-9)
 \end{aligned}$$

The second order relationships of the rotations are :

$$\begin{aligned}
 \ddot{\underline{\varepsilon}} &= \sum_{i=1}^{\text{dof}} \left[\underline{g}_i^R \ddot{\varphi}_i \right] + \sum_{i=1}^{\text{dof}} \sum_{j=1}^{\text{dof}} \left\{ \partial/\partial \varphi_j \left[\underline{g}_i^R \right] \dot{\varphi}_i \dot{\varphi}_j \right\} \\
 &= \sum_{i=1}^{\text{dof}} \left[\underline{g}_i^R \ddot{\varphi}_i \right] + \sum_{i=1}^{\text{dof}} \sum_{j=1}^{\text{dof}} \left\{ \underline{h}_{ij}^R \dot{\varphi}_i \dot{\varphi}_j \right\} \quad (2.2-10)
 \end{aligned}$$

The second order kinematic influence coefficient is defined by the vector derivative of the first order kinematic influence coefficient. This relationship defines all centrifugal and coriolis acceleration terms which result from coupling between velocities due to the rotation of the local reference frames.

2.2.3.1 Second Order Translations

The translational second-order kinematic relationship for a point P at reference position \underline{x}_k on link k is defined by the following:

IF the motions i and j are rotational and occur between the global frame h and point P, AND $k \geq h$:

$${}^{(h)}\underline{h}_{ij}^P(\underline{x}_k) = {}^{(h)}\hat{\underline{S}}_1 \times {}^{(h)}\underline{g}_m^P(\underline{x}_k) \quad (2.2-11a)$$

(where l is the minimum of i and j, and m is the maximum)

IF the motion i and j are rotational and occur between the global frame h and point P, AND $k < h$:

$${}^{(h)}\underline{h}_{ij}^P(\underline{x}_k) = -{}^{(h)}\hat{\underline{S}}_1 \times {}^{(h)}\underline{g}_m^P(\underline{x}_k). \quad (2.2-11b)$$

(where l is the minimum of i and j, and m is the maximum)

IF the motion i is rotational and the motion j is translational, and i occurs between frame h and motion j, and AND $k \geq h$:

$${}^{(h)}\underline{h}_{ij}^P(\underline{x}_k) = {}^{(h)}\hat{\underline{S}}_1 \times {}^{(h)}\underline{g}_j^P(\underline{x}_k) \quad (2.2-11c)$$

IF the motion i is rotational and the motion j is translational, and i occurs between frame h and motion j, and AND $k < h$:

$${}^{(h)}\underline{h}_{ij}^P(\underline{x}_k) = -{}^{(h)}\hat{\underline{S}}_1 \times {}^{(h)}\underline{g}_j^P(\underline{x}_k) \quad (2.2-11d)$$

For all other cases,

$${}^{(h)}\underline{h}_{1j}^P(\underline{x}_k) = \underline{0} \quad (2.2-11e)$$

Note that ${}^{(h)}\hat{\underline{S}}_1$ is always associated with a rotational motion, and ${}^{(h)}\underline{g}_j^P$ may be associated with a rotational motion contributing to the translation of point P, or associated with a translational motion.

2.2.3.2 Second Order Rotations

The rotational second-order kinematic influence coefficient for an element at reference position \underline{x}_k on link k is defined by the following:

IF the motion i is rotational and occurs between the local frame h and motion j, or with motion j, AND $k \geq h$:

$${}^{(h)}\underline{h}_{1j}^R(\underline{x}_k) = {}^{(h)}\hat{\underline{S}}_1 \times {}^{(h)}\underline{g}_j^R(\underline{x}_k) \quad (2.2-12a)$$

IF the motion i is rotational and occurs between the global frame h and motion j, or with motion j, AND $k < h$:

$${}^{(h)}\underline{h}_{1j}^R(\underline{x}_k) = -{}^{(h)}\hat{\underline{S}}_1 \times {}^{(h)}\underline{g}_j^R(\underline{x}_k). \quad (2.2-12b)$$

For all other cases,

$${}^{(h)}h_{1j}^R(\underline{x}_k) = \underline{0} \quad (2.2-12c)$$

Note that in this case, ${}^{(h)}\hat{\underline{S}}_1$ and ${}^{(h)}\underline{g}_j$ are always associated with rotational motions.

2.2.3.3 Second Order Kinematic Effects Due to Gross Motion

The gross rotation of the system can couple with the relative rotations and translations to cause coriolis and centrifugal accelerations, in the same way a relative motion does. The gross translations do not contribute to the the second order accelerations of the system. These kinematic formulas used to compute these kinematic effects are the same as those for the joints, noting that the gross motion occurs at the global frame, and therefore equations 2.2-11a, 11c, and 12a are used for all $0 \leq k \leq n$.

2.2.3.4 Matrix Notation and the Hessian Matrix

The second order kinematic influence coefficients for a point P can be organized into matrix form, called the Hessian matrix. The acceleration can be written as:

$${}^{(h)}\ddot{\underline{R}}(\underline{x}_1) = \left[{}^{(h)}G^P(\underline{x}_1) \right] \ddot{\underline{\varphi}} + \dot{\underline{\varphi}}^T \left[{}^{(h)}H^P(\underline{x}_1) \right] \dot{\underline{\varphi}} \quad (2.2-13)$$

and

$${}^{(h)}\dot{\underline{\omega}}_1(\underline{x}_1) = \left[{}^{(h)}G^R(\underline{x}_1) \right] \ddot{\underline{\varphi}} + \dot{\underline{\varphi}}^T \left[{}^{(h)}H^R(\underline{x}_1) \right] \dot{\underline{\varphi}} \quad (2.2-14)$$

In a more compact notation,

$$\left\{ \begin{array}{l} {}^{(h)}\ddot{\underline{R}}_1(\underline{x}_1) \\ {}^{(h)}\dot{\underline{\omega}}_1(\underline{x}_1) \end{array} \right\} = \left[{}^{(h)}G(\underline{x}_1) \right] \ddot{\underline{\varphi}} + \dot{\underline{\varphi}}^T \left[{}^{(h)}H(\underline{x}_1) \right] \dot{\underline{\varphi}} \quad (2.2-15)$$

The dimension of $\left[{}^{(h)}H(\underline{x}_1) \right]$ is (dof x 6 x dof).

2.2.4 Local Link Kinematics

The velocity of a point on a compliant link, relative to the local reference frame, is defined by a vector function of the mode shapes and the time rates of change of the corresponding generalized coordinates. This information describes the rotational and translational deflections in the

three orthogonal directions. The mode shapes may be a function, or a set of data obtained via modal analysis or a finite element simulation of link deflections. For a lumped parameter model, the mode shapes are assumed to be linear functions. The direction of the motion due to vibrations between reference frames 1 and i+1 is defined by the lines of action in the three orthogonal directions, $\hat{\underline{S}}_1^x$, $\hat{\underline{S}}_1^y$, and $\hat{\underline{S}}_1^z$. Note that these motions take place at the distal end of the link, coinciding with frame i+1, but are measured in local frame 1.

2.2.4.1 Local Link Velocities and Accelerations

The first order relationship which defines the spatial translational and rotational deflectional velocities of a point associated with the reference position ${}^{(1)}\underline{x}_1$, is :

$${}^{(1)}\dot{\underline{d}}_1(\underline{x}_1) = \begin{bmatrix} {}^{(1)}\delta(\underline{x}_1) \end{bmatrix} \dot{\underline{g}}_1 \quad (2.2-16)$$

$${}^{(1)}\dot{\underline{\theta}}_1(\underline{x}_1) = \begin{bmatrix} {}^{(1)}\psi(\underline{x}_1) \end{bmatrix} \dot{\underline{g}}_1 \quad (2.2-17)$$

This simple relationship results from the assumption of the vibrations model that the mode shapes are time invariant. The

form of the second order relationship is similar to the first order relationship:

$${}^{(1)}\ddot{\underline{d}}_1(\underline{x}_1) = \left[{}^{(1)}\underline{\delta}_j(\underline{x}_1) \right] \ddot{\underline{q}}_1 \quad (2.2-18)$$

$${}^{(1)}\ddot{\underline{\theta}}_1(\underline{x}_1) = \left[{}^{(1)}\underline{\psi}_j(\underline{x}_1) \right] \ddot{\underline{q}}_1 \quad (2.2-19)$$

The velocity of the end of the link is the velocity of the point at ${}^{(1)}\underline{x}_1 = {}^{(1)}\underline{L}_1$. Note that the kinematic influence of the vibrational motions are represented as three translations and three rotations at the distal end of the link, but m_1 degrees of freedom are needed to describe the dynamics of these translations and rotations. The velocities of the spatial deflections contribute to the absolute velocities via the appropriate kinematic influence coefficients, which were defined in the previous sections.

2.2.5 Kinematics of the Center of Mass of the System

It is not necessary to formulate the dynamics of the system in terms of the center of mass of the system, although it may be done. It is the opinion of the author that such formulations are unnecessary for the general model of dynamics, and only make the problem more complicated and computational inefficient. The only kinematic information needed to complete this kinematic model is the velocity of the center of mass of the system, relative to the global frame. Note that this

information is not required for the dynamics model, but may be needed for control purposes. Defining μ_j to be the ratio of the mass of link i to the mass of the total system, and \underline{cm}_j to be the vector from the origin of frame j to the center of mass of frame j . The kinematic effect of the rotational motion ϕ_1 on the translational velocity of the center of mass is:

$$\begin{aligned} \dot{\underline{g}}_1^P(R_{-cm}) &= \dot{\underline{S}}_1 \times \left\{ \sum_{j=1}^n \left\{ \mu_j \underline{cm}_j \right\} + \sum_{j=0}^n \left\{ \mu_j \sum_{k=1}^{j-1} \underline{L}_k \right\} \right\} \quad \text{for } i > h \\ \dot{\underline{g}}_1^P(R_{-cm}) &= \dot{\underline{S}}_1 \times \left\{ \sum_{j=0}^{i-1} \left\{ \mu_j \underline{cm}_j \right\} - \sum_{j=0}^{i-1} \left\{ \mu_j \sum_{k=j}^{i-1} \underline{L}_k \right\} \right\} \quad \text{for } i \leq h \end{aligned}$$

(2.2-20)

These kinematic parameters give the translational velocity of the center of mass of the system relative to the origin of the global frame.

2.2.6 Kinematic Models and Momentum Conservation

If no external loads are applied to a system with a base which is free to move relative to the inertial frame, conservation of momentum may be used to derive a first order model of the system, based on the kinematics of the system, and the mass ratios of each link. A system of control based on these equations may be formulated. It can be shown that there is no obvious advantage to computing these equations using the center

of mass formulation, since the necessary equations of the momentum of the system can be obtained from the original kinematics. The reader with an intimate knowledge of dynamics should note that momentum of the system must be calculated with respect to the inertial reference frame. Any computation of the motion of a point in space or on the mechanism, based on the equations of momentum of the system, does not require that the motion of the center of mass be directly computed as an intermediate step. Therefore, any control scheme based on conservation of momentum does not require kinematic formulation in terms of the center of mass of the system.

2.3 The Dynamics of the System

A complete model of system dynamics includes the coupling between gross motion and vibrations. Dynamic coupling terms originate from the kinematics of a system. By measuring joint angles relative to the preceeding link, off-diagonal terms are produced in the rotational jacobian, and in the global inertia modeling matrix for the system. Dynamic coupling will occur in all systems where states (ie. displacement, velocity, ...) are not measured directly from the inertial reference frame.

One of the most interesting, yet subtle examples of coupling is that between the vibrational modes of a link and the gross motions of the link. Most models ignore this coupling by assuming that the off-diagonal terms of the local link inertia matrix, which correspond to this coupling, are zero. These coupling terms, and all other inertia terms, can be derived from expressions of the systems kinetic and potential energies. Lagrange's Equation is then used to produce the equations of motion, resulting in one equation for each degree of freedom of the system. For a system in space (unconstrained), there are six degrees of freedom for the gross motion, one degree of freedom for each joint, and one degree of freedom for each mode of vibration.

2.3.1 Vibration Model Concepts

Vibrations in the system are described by the normal modes, which exhibit harmonic motion at the corresponding root frequency. In theory, there is an infinite number of modes for a continuous body, such as a beam. But in a real system, this theory is not totally accurate in predicting the actual modes, because the assumptions of the theory are only approximations of reality (ie. the beam is not a true continuous system, and material imperfections are not modeled). Also, higher modes usually do not have a measureable effect on the dynamics of the real system, because any large amplitude vibrations at higher frequencies die out quickly due to structural damping. Therefore, the inertia and stiffness terms used in the system model should be based on experimental data obtained from modal analysis, and metrology, or predicted from computational methods such as finite element analysis.

The equivalent mass and stiffness of a link can be obtained by experiments in modal analysis, and other forms of metrology. But unfortunately, these methods do not reveal the inertia terms which describe the coupling between the gross motions and vibrations in the link, which are off-diagonal terms in the inertia matrix. These terms can be derived from the kinetic energy of the system, and a method of predicting their

magnitudes is presented in this study.

To predict the inertia terms of the system, a mode summation method is used where the mode shapes and mass distribution of each link are assumed to be known. This information must be obtained via experimental or computational methods. An accurate knowledge of this information assures that the dynamic model will be accurate. These mode shapes can be represented by any function or set of data. A finite number of modes are used. Lumped parameter models are considered to be a special case, where the mode shapes are simple linear functions and all mass is lumped at the center of mass of the link. To accomodate more flexible links, a link model similar to component mode sythesis is used, allowing the link to be subdivided into smaller sections, or link segments. The rotational and translational deflections of a sub-link are assumed to obey the magnitude constraints imposed by the geometric model of rotations.

There is always a question about when to use lumped parameters, mode summation, or component mode synthesis. No absolute guidelines can be presented as to when each method should be used. As a rule of thumb, lumped parameters should be used to model joint flexibilities and short, fat links. Mode summation should be used to model longer links which exhibit

angular deflections of less than five degrees. Component mode synthesis should be used to model flexible links which have angular deflections greater than five degrees. Note that these models may be "mixed" together in the overall system model.

A firm understanding of the assumptions inherent in these modeling methods is a prerequisite to insuring that the model represents the actual system. It must be stressed that mode summation methods require the beam to be relatively long and thin, but the cross sectional area is not required to be constant. All models of vibrating systems are approximations of very complex phenomena. A good model will not oversimplify the problem. On the other hand, a model should not include more information than is necessary, due to the computational burden imposed. Most importantly, it must be realized that the quality of the information used will have a great effect on the accuracy of model. A qualitative knowledge of the relationship between accuracy of the data used in the model to the accuracy of the results obtained is absolutely necessary.

2.3.2 Kinetic Energy and the System Inertia Matrix

The total kinetic energy of the system is the sum of the kinetic energies of its sub-systems. Each sub-system can be a set of links, one link, or part of a link. Although this is not required, it is convenient to choose each sub-system so the local flexibility matrix is constant. Therefore, for lumped parameter models and mode summation models, a sub-system will be one link, and for component mode synthesis models, a sub-system will be a link segment. The kinetic energy of an element of mass, δm , in subsystem i will be

$$\delta KE_i = \frac{1}{2} \delta m \left\{ \dot{\underline{R}}(\underline{x}_i) \right\} \cdot \left\{ \dot{\underline{R}}(\underline{x}_i) \right\} \quad (2.3-1)$$

where $\dot{\underline{R}}(\underline{x}_i)$ is the absolute velocity of the element on link i . The total kinetic energy for the sub-system is the sum of the kinetic energies of all mass elements. Assuming the mass distribution of the sub-system is known, the kinetic energy can ideally be expressed as a volumetric integral:

$$KE_i = \frac{1}{2} \int_{V_i} \rho(\underline{x}_i) \left\{ \dot{\underline{R}}(\underline{x}_i) \right\} \cdot \left\{ \dot{\underline{R}}(\underline{x}_i) \right\} dV_i \quad (2.3-2)$$

$\dot{\underline{R}}(\underline{x}_i)$ can be expressed as the velocity of a reference point plus the velocity relative to this reference point. By, choosing the reference point to coincide with the local reference frame

origin of the link, the inertia terms commonly associated with the gross motion of the link are obtained, along with the equivalent masses of vibration, and the inertia coupling terms between the gross motion and vibrations. The detailed derivation of the kinetic energy and the inertia terms is presented in Appendix A.

The kinetic energy of the link can be rewritten in terms of the local inertia matrix, the gross motions, and the velocities of vibration for the link.

$$KE_1 = \frac{1}{2} \dot{\phi}_1^T [{}^{(1)}I_1] \dot{\phi}_1$$

$$\text{where } \dot{\phi}_1 = \begin{Bmatrix} {}^{(1)}\dot{\underline{R}}(\underline{Q}_1) \\ {}^{(1)}\dot{\underline{\omega}}(\underline{Q}_1) \\ \dot{\underline{q}}_1 \end{Bmatrix} \quad (2.3-3)$$

and ${}^{(1)}\dot{\underline{R}}(\underline{Q}_1)$ is the translational velocity of the reference frame, ${}^{(1)}\dot{\underline{\omega}}(\underline{Q}_1)$ is the rotational velocity of the reference frame point, and $\dot{\underline{q}}_1$ are the generalized velocities of vibration of link 1.

The local velocities can be expressed in terms of the global reference frame, h , by multiplying by the rotation matrix.

$$\begin{aligned}
 {}^{(h)}\dot{\underline{\phi}}_1 &= \begin{Bmatrix} {}^{(h)}\dot{\underline{R}}(\underline{Q}_1) \\ {}^{(h)}\dot{\underline{\omega}}(\underline{Q}_1) \\ \dot{\underline{q}}_1 \end{Bmatrix} = \begin{Bmatrix} [{}^hT_1] {}^{(1)}\dot{\underline{R}}(\underline{Q}_1) \\ [{}^hT_1] {}^{(1)}\dot{\underline{\omega}}(\underline{Q}_1) \\ \dot{\underline{q}}_1 \end{Bmatrix} \\
 &= \begin{bmatrix} [{}^hT_1] & 0 & 0 \\ 0 & [{}^hT_1] & 0 \\ 0 & 0 & [I] \end{bmatrix} \begin{Bmatrix} {}^{(1)}\dot{\underline{R}}(\underline{Q}_1) \\ {}^{(1)}\dot{\underline{\omega}}(\underline{Q}_1) \\ \dot{\underline{q}}_1 \end{Bmatrix} \quad (2.3-4) \\
 &= [{}^h\mathcal{T}_1] {}^{(1)}\dot{\underline{\phi}}_1
 \end{aligned}$$

$[I]$ is the $m_1 \times m_1$ identity matrix, and $[{}^h\mathcal{T}_1]$ is the augmented transformation matrix. Rewriting the kinetic energy,

$$\begin{aligned}
 KE_1 &= \frac{1}{2} {}^{(1)}\dot{\underline{\phi}}_1^T [{}^{(1)}I_1] {}^{(1)}\dot{\underline{\phi}}_1 \\
 &= \frac{1}{2} {}^{(h)}\dot{\underline{\phi}}_1^T [{}^h\mathcal{T}_1] [{}^{(1)}I_1] [{}^h\mathcal{T}_1]^T {}^{(h)}\dot{\underline{\phi}}_1 \\
 &= \frac{1}{2} {}^{(h)}\dot{\underline{\phi}}_1^T [{}^{(h)}I_1] {}^{(h)}\dot{\underline{\phi}}_1. \quad (2.3-5)
 \end{aligned}$$

At this point, it is necessary to express the kinetic energy in terms of another set of states which consists of the gross motion of the entire system, the velocities of the actuated inputs, and the velocities of all vibrations (ie. at the joints, and in the links) in the system, $\dot{\psi}$. The kinematic relationship between $\dot{\psi}$ and $^{(h)}\dot{\phi}_1$ is:

$$^{(h)}\dot{\phi}_1 = \left[\partial^{(h)}\phi_1 / \partial \psi \right] \dot{\psi} = \left[{}^{(h)}\mathcal{G}_1 \right] \dot{\psi} \quad (2.3-6)$$

where $\left[{}^{(h)}\mathcal{G}_1 \right]$ is the Jacobian for the origin of local coordinate system 1 augmented by an identity relationship which specifies which generalized velocities of vibration are associated with this link:

$$\left[{}^{(h)}\mathcal{G}_1 \right] = \left[\begin{array}{c} {}^{(h)}G_1(0) \\ \partial \mathcal{G}_1 / \partial \psi \end{array} \right] \quad (2.3-7)$$

Substituting this expression into the kinetic energy,

$$\begin{aligned} KE_1 &= \frac{1}{2} \dot{\psi}^T \left[{}^{(h)}\mathcal{G}_1 \right]^T \left[{}^{(h)}\mathcal{G}_1 \right] \left[{}^{(1)}I_1 \right] \left[{}^{(h)}\mathcal{G}_1 \right]^T \left[{}^{(h)}\mathcal{G}_1 \right] \dot{\psi}^{(h)} \\ &= \frac{1}{2} \dot{\psi}^T \left[{}^{(h)}\mathcal{G}_1 \right]^T \left[{}^{(h)}I_1 \right] \left[{}^{(h)}\mathcal{G}_1 \right] \dot{\psi} \\ &= \frac{1}{2} \dot{\psi}^T \left[{}^{(h)}I_1^* \right] \dot{\psi} \end{aligned} \quad (2.3-8)$$

where $\left[{}^{(h)}I_1^* \right]$ the the inertia modeling matrix for sub-system

i. The total kinetic energy of the system is the sum of the kinetic energy of each sub-system

$$\begin{aligned}
 KE_{\text{system}} &= \sum_{i=1}^{n+1} KE_i \\
 &= \sum_{i=1}^{n+1} \frac{1}{2} \dot{\varphi}^T [{}^{(h)}I_i^*] \dot{\varphi} \\
 &= \frac{1}{2} \dot{\varphi}^T [{}^{(h)}I^*] \dot{\varphi}
 \end{aligned} \tag{2.3-9}$$

where $[{}^{(h)}I^*]$ is the inertia modeling matrix of the entire system.

2.3.3 Potential Energy and the Stiffness Matrix

The potential energy due to compliance in the system is usually a linear function of the generalized coordinates of vibration. The potential energy of a differential element of link 1 can be expressed as

$$\delta PE_1 = \frac{1}{2} \delta(EI) \left\{ \underline{d}(\underline{x}_1) \right\} \cdot \left\{ \underline{d}(\underline{x}_1) \right\}, \quad (2.3-10)$$

and ideally may be represented as a volumetric integral over the link

$$PE_1 = \frac{1}{2} \int_{V_1} EI/V(\underline{x}_1) \left\{ \underline{d}(\underline{x}_1) \right\} \cdot \left\{ \underline{d}(\underline{x}_1) \right\} dV_1. \quad (2.3-11)$$

A detailed derivation of the potential energy and stiffness terms is presented in Appendix B.

The potential energy of the link can be rewritten in terms of the local stiffness matrix and the generalized coordinates of vibration for the link:

$$PE_1 = \underline{q}_1^T [K_1] \underline{q}_1. \quad (2.3-12)$$

Noting that the flexibilities may be extracted from the state position vector via an identity relationship defined as

$$q_1 = \left[\partial q_1 / \partial \varphi \right] \varphi. \quad (2.3-13)$$

the potential energy can be written in terms of the system state vector,

$$\begin{aligned} PE_1 &= \varphi^T \left[\partial q_1 / \partial \varphi \right]^T [K_1] \left[\partial q_1 / \partial \varphi \right] \varphi \\ &= \varphi^T [K_1^*] \varphi \end{aligned} \quad (2.3-14)$$

where $[K_1^*]$ is the flexibility modeling matrix of link 1. The total potential energy of the system is the sum of the potential energy of the links.

$$\begin{aligned} PE_{\text{system}} &= \sum_{i=1}^{n+1} PE_i \\ &= \sum_{i=1}^{n+1} \frac{1}{2} \varphi^T [K_i^*] \varphi \\ &= \frac{1}{2} \varphi^T [K^*] \varphi \end{aligned} \quad (2.3-15)$$

where $[K^*]$ is the flexibility modeling matrix of the entire system.

2.3.4 The Equations of Motion and the Power Modeling Matrix

The equations of dynamics are obtained via Lagrange's equation. For a multi-degree of freedom system, it can be expressed as:

$$\underline{\tau} = d/dt \left[\partial KE / \partial \dot{\underline{\varphi}}^T \right] - \partial KE / \partial \underline{\varphi}^T + \partial PE / \partial \underline{\varphi}^T . \quad (2.3-16)$$

The term $\partial KE / \partial \dot{\underline{\varphi}}^T$ can be readily obtained due to the quadratic form of KE.

$$\begin{aligned} \partial KE / \partial \dot{\underline{\varphi}}^T &= \frac{1}{2} \left[{}^{(h)}\mathbf{I}^* \right] \dot{\underline{\varphi}} + \frac{1}{2} \left[{}^{(h)}\mathbf{I}^* \right]^T \dot{\underline{\varphi}} + \frac{1}{2} \dot{\underline{\varphi}}^T \partial / \partial \dot{\underline{\varphi}}^T \left[{}^{(h)}\mathbf{I}^* \right] \dot{\underline{\varphi}} \\ &= \left[{}^{(h)}\mathbf{I}^* \right] \dot{\underline{\varphi}} \end{aligned} \quad (2.3-17)$$

noting that $\left[{}^{(h)}\mathbf{I}^* \right]$ is symmetric, and not a function of velocity. The time derivative of this term is

$$\begin{aligned} d/dt \left[\partial KE / \partial \dot{\underline{\varphi}}^T \right] &= d/dt \left\{ \left[{}^{(h)}\mathbf{I}^* \right] \dot{\underline{\varphi}} \right\} \\ &= \left\{ d/dt \left[{}^{(h)}\mathbf{I}^* \right] \right\} \dot{\underline{\varphi}} + \left[{}^{(h)}\mathbf{I}^* \right] \ddot{\underline{\varphi}} . \end{aligned} \quad (2.3-18)$$

The time derivative of the inertia modeling matrix can be written as

$$\frac{d}{dt} [{}^{(h)}I^*] = \dot{\varphi}^T \left\{ \frac{d}{d\varphi} [{}^{(h)}I^*] \right\}^{PT}, \quad (2.3-19)$$

where the planar transpose operator performs the following operation:

$$\left\{ \frac{d}{d\varphi} [{}^{(h)}I^*] \right\}_{i;j;k}^{PT} = \left\{ \frac{d}{d\varphi} [{}^{(h)}I^*] \right\}_{j;i;k} \quad (2.3-20)$$

for all i , j , and k , where the notation is plane;row;column.

The next term of Lagrange's Equation is

$$\begin{aligned} \partial KE / \partial \varphi^T &= \partial / \partial \varphi^T \left\{ \frac{1}{2} \dot{\varphi}^T [{}^{(h)}I^*] \dot{\varphi} \right\} \\ &= \frac{1}{2} \dot{\varphi}^T \left\{ \partial / \partial \varphi [{}^{(h)}I^*] \right\} \dot{\varphi}. \end{aligned} \quad (2.3-21)$$

The last term yields

$$\begin{aligned} \partial PE / \partial \varphi^T &= \partial PE / \partial \varphi^T \left\{ \frac{1}{2} \varphi^T [K^*] \varphi \right\} \\ &= [K^*] \varphi. \end{aligned} \quad (2.3-22)$$

The system of equations can be written as

$$\begin{aligned} \underline{\tau} = [^{(h)}I^*] \ddot{\varphi} + \dot{\varphi}^T \left[\left\{ \partial / \partial \varphi [^{(h)}I^*] \right\}^{PT} - \frac{1}{2} \left\{ \partial / \partial \varphi [^{(h)}I^*] \right\} \right] \dot{\varphi} \\ + [K^*] \varphi . \end{aligned} \quad (2.3-23)$$

These terms can be written in a more compact matrix form as a matrix with dimensions (dof x dof x dof):

$$[^{(h)}P^*] = \left\{ \partial / \partial \varphi [^{(h)}I^*] \right\}^{PT} - \frac{1}{2} \left\{ \partial / \partial \varphi [^{(h)}I^*] \right\}, \quad (2.3-24)$$

so the dynamic equations can be written in a standard matrix notation,

$$\underline{\tau} = [^{(h)}I^*] \ddot{\varphi} + \dot{\varphi}^T [^{(h)}P^*] \dot{\varphi} + [K^*] \varphi . \quad (2.3-25)$$

Recalling the expression for $[^{(h)}I^*]$,

$$[^{(h)}I^*] = \sum_{i=1}^{n+1} [^{(h)}\mathcal{G}_i]^T [^{(h)}I_i] [^{(h)}\mathcal{G}_i] , \quad (2.3-26)$$

The term $\frac{1}{2} \partial/\partial\varphi [^{(h)}I^*]$ can be expressed as

$$\begin{aligned}
 &= \sum_{i=1}^{n+1} \frac{1}{2} d/d\varphi \left\{ \left[^{(h)}\mathcal{G}_i \right]^T \left[^{(h)}I_i \right] \left[^{(h)}\mathcal{G}_i \right] \right\} \\
 &= \sum_{i=1}^{n+1} \frac{1}{2} d/d\varphi \left\{ \left[^{(h)}\mathcal{G}_i \right]^T \right\} \left[^{(h)}I_i \right] \left[^{(h)}\mathcal{G}_i \right] \\
 &\quad + \sum_{i=1}^{n+1} \frac{1}{2} \left[^{(h)}\mathcal{G}_i \right]^T d/d\varphi \left\{ \left[^{(h)}I_i \right] \right\} \left[^{(h)}\mathcal{G}_i \right] \\
 &\quad + \sum_{i=1}^{n+1} \frac{1}{2} \left[^{(h)}\mathcal{G}_i \right]^T \left[^{(h)}I_i \right] d/d\varphi \left\{ \left[^{(h)}\mathcal{G}_i \right] \right\} \quad (2.3-27)
 \end{aligned}$$

Notice that the second order kinematic influence coefficients are defined by

$$d/d\varphi \left[^{(h)}\mathcal{G}_i \right] = \left[^{(h)}\mathcal{H}_i \right]. \quad (2.3-28)$$

Substituting the local inertia matrix referenced to the local frame,

$$\begin{aligned}
 \partial/\partial\varphi \left[^{(h)}I_i \right] &= \partial/\partial\varphi \left\{ \left[^{h}\mathcal{G}_i \right] \left[^{(1)}I_i \right] \left[^{h}\mathcal{G}_i \right]^T \right\} \\
 &= \partial/\partial\varphi \left\{ \left[^{h}\mathcal{G}_i \right] \right\} \left[^{(1)}I_i \right] \left[^{h}\mathcal{G}_i \right]^T \\
 &\quad + \left[^{h}\mathcal{G}_i \right] \partial/\partial\varphi \left\{ \left[^{(1)}I_i \right] \right\} \left[^{h}\mathcal{G}_i \right]^T \\
 &\quad + \left[^{h}\mathcal{G}_i \right] \left[^{(1)}I_i \right] \partial/\partial\varphi \left\{ \left[^{h}\mathcal{G}_i \right]^T \right\} \quad (2.3-29)
 \end{aligned}$$

where $\partial/\partial\varphi \left\{ \left[{}^{(1)}I_1 \right] \right\}$ will be denoted henceforth as $[\partial {}^{(1)}I_1/\partial\varphi]$.

Recalling the exact form of the augmented transformation matrix,

$$\partial/\partial\varphi \left\{ \left[{}^hT_1 \right] \right\} = \partial/\partial\varphi \begin{bmatrix} [{}^hT_1] & 0 & 0 \\ 0 & [{}^hT_1] & 0 \\ 0 & 0 & [I] \end{bmatrix} \quad (2.3-30)$$

The columns of the transformation matrix, $[{}^hT_1]$, are made up of the direction cosines of the i th local coordinate frame. These direction cosines are a function of the rotational displacements between the global reference frame and the local frame. This kinematic relationship can be expressed by the cross-product of the first order rotational influence coefficients and the direction cosines of the local i th frame. This results in the second-order relationship:

$$\partial/\partial\varphi_j [{}^hT_1] = \quad (2.3-31)$$

$$\left[[{}^{(h)}G_1^{jk}]_{;j} x[{}^hT_1]_{;1} \mid [{}^{(h)}G_1^{jk}]_{;j} x[{}^hT_1]_{;2} \mid [{}^{(h)}G_1^{jk}]_{;j} x[{}^hT_1]_{;3} \right]$$

The skew-symmetric form of $[(h)G_1^{jk}]$ may be substituted for the cross product, yielding

$$\partial/\partial\varphi_j [h_{T_1}] = \quad (2.3-32)$$

$$\left[[(h)\tilde{G}_1^{jk}]_{;j} [h_{T_1}]_{;1} + [(h)\tilde{G}_1^{jk}]_{;j} [h_{T_1}]_{;2} + [(h)\tilde{G}_1^{jk}]_{;j} [h_{T_1}]_{;3} \right]$$

or, in more compact notation,

$$\partial/\partial\varphi_j [h_{T_1}] = [(h)\tilde{G}_1^{jk}]_{;j} [h_{T_1}] . \quad (2.3-33)$$

Now, the term $\partial/\partial\varphi_j \left\{ \left[h_{\mathcal{T}_1} \right] \right\}$ can be expressed as

$$\partial/\partial\varphi_j \left\{ \left[h_{\mathcal{T}_1} \right] \right\} = \begin{bmatrix} [(h)\tilde{G}_1^{jk}]_{;j} & 0 & 0 \\ 0 & [(h)\tilde{G}_1^{jk}]_{;j} & 0 \\ 0 & 0 & 0 \end{bmatrix} \left[h_{\mathcal{T}_1} \right] . \quad (2.3-34)$$

Letting the matrix of skew-symmetric forms be

$$\begin{aligned}
 \left[\begin{smallmatrix} (h) \tilde{g}_1^{jk} \end{smallmatrix} \right] &= \begin{bmatrix} \left[\begin{smallmatrix} (h) \tilde{G}_1^{jk} \end{smallmatrix} \right]_{;1} & 0 & 0 \\ 0 & \left[\begin{smallmatrix} (h) \tilde{G}_1^{jk} \end{smallmatrix} \right]_{;1} & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
 &\quad \begin{bmatrix} \left[\begin{smallmatrix} (h) \tilde{G}_1^{jk} \end{smallmatrix} \right]_{;2} & 0 & 0 \\ 0 & \left[\begin{smallmatrix} (h) \tilde{G}_1^{jk} \end{smallmatrix} \right]_{;2} & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
 &\quad \vdots \\
 &\quad \begin{bmatrix} \left[\begin{smallmatrix} (h) \tilde{G}_1^{jk} \end{smallmatrix} \right]_{;n} & 0 & 0 \\ 0 & \left[\begin{smallmatrix} (h) \tilde{G}_1^{jk} \end{smallmatrix} \right]_{;n} & 0 \\ 0 & 0 & 0 \end{bmatrix}
 \end{bmatrix} \quad (2.3-35)
 \end{aligned}$$

the derivative in question can be expressed in a much more compact form:

$$\partial / \partial \varphi \left\{ \left[\begin{smallmatrix} h g_1 \end{smallmatrix} \right] \right\} = \left[\begin{smallmatrix} (h) \tilde{g}_1^{jk} \end{smallmatrix} \right] \left[\begin{smallmatrix} h g_1 \end{smallmatrix} \right]. \quad (2.3-36)$$

The final resulting equation for the term $\partial/\partial\varphi [^{(h)}I^*]$ is a function of the local inertia tensor matrix and the geometry of the system:

$$\begin{aligned} \partial/\partial\varphi [^{(h)}I^*] = \sum_{i=1}^{n+1} \left\{ \right. & \\ & [^{(h)}\mathcal{H}_i]^T [{}^{hg}_i] [^{(h)}I_i] [{}^{hg}_i]^T [^{(h)}\mathcal{G}_i] \\ & + [^{(h)}\mathcal{G}_i]^T [{}^{hg}_i] [^{(h)}I_i] [{}^{hg}_i]^T [^{(h)}\mathcal{H}_i] \\ & + [^{(h)}\mathcal{G}_i]^T [^{(h)}\tilde{\mathcal{G}}_i^{jk}] [{}^{hg}_i] [^{(h)}I_i] [{}^{hg}_i]^T [^{(h)}\mathcal{G}_i] \\ & + [^{(h)}\mathcal{G}_i]^T [{}^{hg}_i] [^{(h)}I_i] [{}^{hg}_i]^T [^{(h)}\tilde{\mathcal{G}}_i^{jk}]^T [^{(h)}\mathcal{G}_i] \\ & \left. + [^{(h)}\mathcal{G}_i]^T [{}^{hg}_i] [\partial^{(i)}I_i/\partial\varphi] [{}^{hg}_i]^T [^{(h)}\mathcal{G}_i] \right\} \quad (2.3-37) \end{aligned}$$

It has dimensions of (dof x dof x dof).

2.3.5 Applied Forces and Torques

The applied torque vector $\underline{\tau}$ is the equivalent load of any point loads or distributed loads which may be applied to the system. The equivalent load which corresponds to an applied point force and point moment at position \underline{x}_i on link i is:

$$\underline{\tau}_{\text{equivalent}} = \left[{}^{(h)}\mathcal{G}(\underline{x}_i) \right]^T \underline{\tau}_{\text{applied}} \quad (2.3-38)$$

The equivalent load expression for a distributed load is

$$\underline{\tau}_{\text{equivalent}} = \sum_{i=0}^n \left\{ \int_V \left[{}^{(h)}\mathcal{G}(\underline{x}_i) \right]^T \delta \underline{\tau}_{\text{applied}} \right\} \quad (2.3-39)$$

which is a volumetric integral of the distributed load over each link, or sub-system. This is particularly important for an unconstrained system in a gravity, or pressure field. For orbiting systems, a slight differential in the gravity gradient field will cause a torque on the system. For underwater systems, a similar effect will occur due to the variation of water pressure with depth.

This gravity gradient effect can be estimated by assuming the distributed load to be a point load at the center of mass. This leads to the equation

$$\tau_{\text{equivalent}} \cong \sum_{i=0}^n \left\{ \left[{}^{(h)}\mathcal{G}(\underline{cm}_i) \right]^T \tau_{\text{approx}_i} \right\}. \quad (2.3-40)$$

This same method can be used to estimate the equivalent loads of an underwater system, systems which develop fluid drag, systems working in a centrifuge, or systems in a gravity field.

Damping terms have not been presented explicitly in this study, but may be modeled as loads which are functions of the magnitude of the vibrations, or the velocity of the vibrations. These loads are assumed to occur "internally", and are written in terms of the states of the system, $\dot{\varphi}$, and φ . Therefore it is not necessary to use the Jacobian to transform the damping loads, as it is with externally applied loads.

3.0 Dynamic Simulations:

The product of this research is a general model of the dynamics of serial manipulators. The model derived has been implemented in a simulation package in "C" (a programming language) which currently resides on a Silicon Graphics 4-D computer system at the University of Texas Mechanical Systems Robotics Laboratory. The source code for this program, called VSim, can be found in Appendix C.

VSim is a general simulation package, which can simulate serial systems of n links. Each link can have m_t translational vibration modes and m_r rotational vibration modes. The number of modes may differ from link to link (ie. link 2 may have 1 translational mode and 5 rotational modes, while link 1 may have 3 translational modes ...). A link may be modeled as a rigid body by assigning it no vibrational modes. Extremely flexible links may be subdivided and modeled as several sub-link. Each sub-link is modeled the same as a regular link. Each sub-link may be modeled with several modes.

Because of the generality of the program, it must be stressed that the output is only as good as the input. If the input data truly represents the real system, there will be a good chance that the output data also represents the real system. The input to VSim includes geometry, inertia, and stiffness data for each link, along with a variety of "bookkeeping" data. In

order to promote efficiency in computations, yet preserve the general capabilities of the program, the current version requires that the input data be included in the file "Robotdat.c". The control model must be included in the file "Model.c" which is accessed by the integration routine. The program generates the equations of motion internally, from the data provided, and from the control algorithm provided. The numerical integration algorithm uses a predictor-corrector method with adaptive step size and adaptive order, which was presented by C. William Gear [24].

3.1 Example: A Large Space-Based Robotic System

As an example, the dynamic equations of a large flexible robotic system used in space operations are formulated, and the resulting system is simulated under various disturbing inputs to the end-effector and joints. The system consists of a large spacecraft with a 55 ft. manipulator which has 6 actuated joints. The system was modeled such that the base link, (the spacecraft) could translate relative to the inertial reference frame. The geometry, mass, and stiffness information used for this example is similar to that of the Remote Manipulator System used by NASA. A truncated mode summation method will be used to model translational vibrations, and a lumped parameter method will be used to rotational vibrations.

Table 3.1-1

Geometric Data for Example Robot:

Link #	\underline{L} (ft)	α (rad)	θ (rad)	joint type
0	$\begin{Bmatrix} -35.88 \\ -8.07 \\ 0.8 \end{Bmatrix}$	π	-	-
1	$\begin{Bmatrix} 0.0 \\ 0.0 \\ 1.0 \end{Bmatrix}$	$\frac{\pi}{2}$	π	revolute
2	$\begin{Bmatrix} 20.92 \\ 0.0 \\ 0.0 \end{Bmatrix}$	0	0	revolute
3	$\begin{Bmatrix} 23.18 \\ 0.0 \\ 0.0 \end{Bmatrix}$	0	0	revolute
4	$\begin{Bmatrix} 1.5 \\ 0.0 \\ 0.0 \end{Bmatrix}$	$3 \frac{\pi}{2}$	0	revolute
5	$\begin{Bmatrix} 0.0 \\ -2.5 \\ 0.0 \end{Bmatrix}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	revolute
6	$\begin{Bmatrix} 0.0 \\ 0.0 \\ 2.17 \end{Bmatrix}$	0	0	revolute

Table 3.1-2

Mass and Inertia for the Example Robot

Link #	mass (slugs)	<u>cm</u> (ft)	<u>I</u> (slug·ft ²)
0	6.39e3	$\begin{Bmatrix} 0.0 \\ 0.0 \\ 0.0 \end{Bmatrix}$	$\begin{bmatrix} 8.9e5 & 1.6e4 & -2.9e5 \\ 1.6e4 & 6.9e6 & 3.5e3 \\ -2.9e5 & 3.5e3 & 7.3e6 \end{bmatrix}$
1	2.198	$\begin{Bmatrix} 0.0 \\ 0.0 \\ 0.5 \end{Bmatrix}$	$\begin{bmatrix} 1.37 & .014 & -.059 \\ .014 & 1.595 & -.027 \\ -.059 & -.027 & .439 \end{bmatrix}$
2	9.536	$\begin{Bmatrix} 10.46 \\ 0.0 \\ 0.0 \end{Bmatrix}$	$\begin{bmatrix} 2.38 & -6.77 & 1.423 \\ -6.77 & 1.8e3 & -.223 \\ 1.423 & -.223 & 1.8e3 \end{bmatrix}$
3	5.982	$\begin{Bmatrix} 11.58 \\ 0.0 \\ 2.17 \end{Bmatrix}$	$\begin{bmatrix} 1.17 & 2.79 & -.592 \\ 2.79 & 8.5e2 & -.059 \\ -.592 & -.059 & 1.3e3 \end{bmatrix}$
4	0.58	$\begin{Bmatrix} 0.75 \\ 0.0 \\ 0.0 \end{Bmatrix}$	$\begin{bmatrix} .066 & 0.0 & 0.0 \\ 0.0 & .622 & 0.0 \\ 0.0 & 0.0 & .622 \end{bmatrix}$
5	3.144	$\begin{Bmatrix} 0.0 \\ -1.25 \\ 0.0 \end{Bmatrix}$	$\begin{bmatrix} 8.36 & 0.0 & 0.0 \\ 0.0 & .301 & 0.0 \\ 0.0 & 0.0 & 8.36 \end{bmatrix}$
6	3.094	$\begin{Bmatrix} 0.0 \\ 0.0 \\ 1.08 \end{Bmatrix}$	$\begin{bmatrix} 5.71 & .261 & -.712 \\ .261 & .301 & 0.0 \\ -.721 & 0.0 & 8.63 \end{bmatrix}$

Links 2 and 3 were modeled as flexible, and the other links were considered to be rigid bodies. The translational vibrations transverse to the length of the link were modeled by cubic modes, and the torsion about the minor axis of the links were modeled by simple linear rotational springs. The inertia terms were computed using the equations derived in Appendix A.

The translational cubic mode shapes are:

$$\delta_{-1}(\underline{x}_1) = \begin{Bmatrix} 0 \\ a_1 + b_1(x_1/L_1) + c_1(x_1/L_1)^2 + d_1(x_1/L_1)^3 \\ 0 \end{Bmatrix} \quad (3.1-1)$$

$$\delta_{-2}(\underline{x}_1) = \begin{Bmatrix} 0 \\ a_2 + b_2(x_1/L_1) + c_2(x_1/L_1)^2 + d_2(x_1/L_1)^3 \\ 0 \end{Bmatrix} \quad (3.1-2)$$

$$\delta_{-3}(\underline{x}_1) = \begin{Bmatrix} 0 \\ 0 \\ a_3 + b_3(x_1/L_1) + c_3(x_1/L_1)^2 + d_3(x_1/L_1)^3 \end{Bmatrix} \quad (3.1-3)$$

and

$$\delta_{-4}(\underline{x}_1) = \begin{Bmatrix} 0 \\ 0 \\ a_4 + b_4(x_1/L_1) + c_4(x_1/L_1)^2 + d_4(x_1/L_1)^3 \end{Bmatrix} \quad (3.1-4)$$

The proximal end of the link is not allowed to deflect relative to the local coordinate system,

$$\underline{\delta}_j(\underline{0}_1) = \underline{\delta}_j(\underline{0}_1) = \left\{ \underline{0} \right\} \quad (3.1-5)$$

therefore $a_j = 0$. The slope of the deflection is also assumed to be zero at the local coordinate system, therefore $b_j = 0$.

The orthogonality condition needed to assure that these modes are independent is expressed as

$$\int_V \underline{\delta}_j(\underline{x}_1) \cdot \underline{\delta}_k(\underline{x}_1) dV = 0 \quad (3.1-6)$$

$$= \int_0^L \left[c_1 c_2 (x_1/L_1)^4 + (c_1 d_2 + c_2 d_1) (x_1/L_1)^5 + d_1 d_2 (x_1/L_1)^6 \right] dx_1. \quad (3.1-7)$$

Integrating, we find the orthogonality requirement to be

$$r_2 = -(42 + 35 r_1) / (35 + 30 r_1) \quad (3.1-8)$$

where $r_1 = d_1/c_1$, and $r_2 = d_2/c_2$. For this example, the first ratio is $r_1 = 0$, making the second ratio, $r_2 = -1.2$.

The mode shape functions relating the local deflections to the generalized coordinates of the local link are:

$$\underline{\delta}_1(\underline{x}_1) = \begin{Bmatrix} 0 \\ (x_1/L_1)^2 \\ 0 \end{Bmatrix} \quad (3.1-9)$$

$$\underline{\delta}_2(\underline{x}_1) = \begin{Bmatrix} 0 \\ (x_1/L_1)^2 - 1.2 (x_1/L_1)^3 \\ 0 \end{Bmatrix} \quad (3.1-10)$$

$$\underline{\delta}_3(\underline{x}_1) = \begin{Bmatrix} 0 \\ 0 \\ (x_1/L_1)^2 \end{Bmatrix} \quad (3.1-11)$$

$$\underline{\delta}_4(\underline{x}_1) = \begin{Bmatrix} 0 \\ 0 \\ (x_1/L_1)^2 - 1.2 (x_1/L_1)^3 \end{Bmatrix} \quad (3.1-12)$$

These mode shapes can be organized into matrix form which gives the local link translational deflections.

$$\underline{d}(\underline{x}_1) = \begin{bmatrix} \underline{\delta}(\underline{x}_1) \end{bmatrix} \dot{\underline{q}}_1 \quad (3.1-13)$$

$$= \begin{bmatrix} \underline{\delta}_1(\underline{x}_1) + \underline{\delta}_2(\underline{x}_1) + \underline{\delta}_3(\underline{x}_1) + \underline{\delta}_4(\underline{x}_1) + \underline{0} \end{bmatrix} \begin{Bmatrix} q_{11} \\ q_{21} \\ q_{31} \\ q_{41} \\ q_{51} \end{Bmatrix}$$

The rotational mode shapes are assumed to be simple linear functions, which is the prescribed assumption for lumped parameter models:

$$\psi_1(\underline{x}_1) = \begin{Bmatrix} x_1/L_1 \\ 0 \\ 0 \end{Bmatrix} \quad (3.1-14)$$

and may be organized into matrix form to obtain the local rotational deflection.

$$\underline{\theta}(\underline{x}_1) = \begin{bmatrix} \underline{0} & | & \underline{0} & | & \underline{0} & | & \underline{0} & | & \psi_1(\underline{x}_1) \end{bmatrix} \begin{Bmatrix} q_{11} \\ q_{21} \\ q_{31} \\ q_{41} \\ q_{51} \end{Bmatrix}. \quad (3.1-15)$$

The shape functions will now be used to compute the coupling terms and vibration terms of the local inertia matrices. From Appendix A, the inertial coupling between the translational flexibilities and the translational motion of the link are

$$\begin{bmatrix} I_{mq_1} \end{bmatrix} = \begin{bmatrix} M_{d11} & | & M_{d12} & | & M_{d13} & | & M_{d14} & | & \underline{0} \end{bmatrix}, \quad (3.1-16)$$

such that

$$M_{d1j} = \int_V \rho(\underline{x}_1) \left[\underline{\delta}_j(\underline{x}_1) \right] dV. \quad (3.1-17)$$

Assuming that the mass of the link is distributed evenly along the length of the link, such that $\mu_1 = m_1/L_1$, the matrix becomes

$$\begin{bmatrix} I_{mq_1} \end{bmatrix} = \mu_1 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ c_1 L_1/3 & c_2 L_1/30 & 0 & 0 & 0 \\ 0 & 0 & c_1 L_1/3 & c_2 L_1/30 & 0 \end{bmatrix} \quad (3.1-18)$$

For link number 2 this matrix is:

$$\begin{bmatrix} I_{mq_2} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 3.18 c_1 & .318 c_2 & 0 & 0 & 0 \\ 0 & 0 & 3.18 c_1 & .318 c_2 & 0 \end{bmatrix} \quad (3.1-19)$$

and for link number 3 this matrix is:

$$\begin{bmatrix} I_{mq_3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1.99 c_1 & .199 c_2 & 0 & 0 & 0 \\ 0 & 0 & 1.99 c_1 & .199 c_2 & 0 \end{bmatrix} \quad (3.1-20)$$

The coefficients c_1 and c_2 are scale factors.

The coupling terms between the gross rotation and the translational flexibility are presented in Appendix A as:

$$\begin{bmatrix} I_{Lq_1}^P \end{bmatrix} = \begin{bmatrix} M_{Ld11}^P & M_{Ld12}^P & M_{Ld13}^P & M_{Ld14}^P & 0 \end{bmatrix} \quad (3.1-21)$$

such that

$$M_{Ld11}^P = \int_V \rho(\underline{x}_1) \begin{bmatrix} \tilde{x}_1 \end{bmatrix} \delta_j(\underline{x}_1) dV. \quad (3.1-22)$$

The resulting matrix is

$$\begin{bmatrix} I_{Lq_1}^P \end{bmatrix} = \mu_1 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -c_1 L_1^2/4 & -c_2 L_1^2/100 & 0 \\ c_1 L_1/4 & c_2 L_1/100 & 0 & 0 & 0 \end{bmatrix} \quad (3.1-22)$$

The coupling between the rotational vibrations and the rotational gross motion was presented in Appendix A.

$$\begin{bmatrix} I_{mq_1}^T \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & M_{Ld15}^T \end{bmatrix} \quad (3.1-23)$$

This matrix becomes

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & I_{xx1}/2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.1-24)$$

By combining these matrices we obtain the total coupling between the vibrations and the rotational gross motion.

For link number 2 this matrix is

$$\begin{bmatrix} I_{Lq_2} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1.19 \\ 0 & 0 & -49.8 c_1 & -1.99 c_2 & 0 \\ 49.8 c_1 & 1.99 c_2 & 0 & 0 & 0 \end{bmatrix}, \quad (3.1-25)$$

and the for link number 3 this matrix is

$$\begin{bmatrix} I_{Lq_3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & .585 \\ 0 & 0 & -34.6 c_1 & -1.39 c_2 & 0 \\ 34.6 c_1 & 1.39 c_2 & 0 & 0 & 0 \end{bmatrix}. \quad (3.1-26)$$

The generalized inertias of vibration are defined in Appendix A as

$$I_{qq_{ijk}}^P = \int_V \rho(\underline{x}_i) \delta_j(\underline{x}_i) \cdot \delta_k(\underline{x}_i) dV, \quad (3.1-27)$$

and for rotational vibrations it is given as

$$I_{qq_{ijk}}^R = \int_V I(\underline{x}_i) \psi_j(\underline{x}_i) \cdot \psi_k(\underline{x}_i) dV. \quad (3.1-28)$$

The results of this integration may be organized into a matrix for each flexible link.

$$\begin{bmatrix} I_{qq_5} \end{bmatrix} = \begin{bmatrix} 2.38 c_1^2 & 0 & 0 & 0 & 0 \\ 0 & .05449 c_2^2 & 0 & 0 & 0 \\ 0 & 0 & 2.38 c_1^2 & 0 & 0 \\ 0 & 0 & 0 & .05449 c_2^2 & 0 \\ 0 & 0 & 0 & 0 & .793 \end{bmatrix}$$

(3.1-29)

$$\begin{bmatrix} I_{qq_5} \end{bmatrix} = \begin{bmatrix} 1.45 c_1^2 & 0 & 0 & 0 & 0 \\ 0 & .03419 c_2^2 & 0 & 0 & 0 \\ 0 & 0 & 1.45 c_1^2 & 0 & 0 \\ 0 & 0 & 0 & .03419 c_2^2 & 0 \\ 0 & 0 & 0 & 0 & .39 \end{bmatrix}$$

(3.1-30)

The stiffness terms for these two links are obtained from the equations presented in Appendix B. The resulting stiffness matrix for link 2 is:

$$\begin{bmatrix} K_2 \end{bmatrix} = \begin{bmatrix} 4.5e3 c_1^2 & -3.5e3 c_1 c_2 & 0 & 0 & 0 \\ -3.5e3 c_1 c_2 & 7.7e3 c_2^2 & 0 & 0 & 0 \\ 0 & 0 & 4.5e3 c_1^2 & -3.5e3 c_1 c_2 & 0 \\ 0 & 0 & -3.5e3 c_1 c_2 & 7.7e3 c_2^2 & 0 \\ 0 & 0 & 0 & 0 & 1.1e6 \end{bmatrix} \quad (3.1-31)$$

and for link 3:

$$\begin{bmatrix} K_3 \end{bmatrix} = \begin{bmatrix} 2.2e3 c_1^2 & -1.7e3 c_1 c_2 & 0 & 0 & 0 \\ -1.7e3 c_1 c_2 & 3.8e3 c_2^2 & 0 & 0 & 0 \\ 0 & 0 & 2.2e3 c_1^2 & -1.7e3 c_1 c_2 & 0 \\ 0 & 0 & -1.7e3 c_1 c_2 & 3.8e3 c_2^2 & 0 \\ 0 & 0 & 0 & 0 & 8.3e5 \end{bmatrix} \quad (3.1-32)$$

This system was simulated with various disturbances applied to the point of resolution (The point of resolution is a point at the end-effector of the manipulator). In the first simulation, a step load of 1000 lb. and 1000 ft-lb. was applied along each axis of the global frame. The simulation was performed once for configuration #1, for which all joint displacements are zero, and once for configuration #2, for which all joints are zero, except for joint 3, which was set at $\pi/2$

radians. The base was allowed to translate. The magnitudes of the first and second translational modes for each flexible link are shown in Figures 3.1-1a through 3.1-4b, for reference position #1. The translational modes are dominated by frequencies around 3 Hz. The impulse response of the rotational mode was found to be much faster than that of the translational modes. This is because of the small inertia of the system in configuration #1. The rotational vibrations of the system near configuration #1 will be dominated by the loadings at the end-effector. If a payload with large inertia is being moved, the frequency of the rotational vibrations will be lower. The simulation for configuration #2, shown in Figure 3.1-5a through 3.1-8b, shows the configuration dependence of the system response. These simulations reveal a much lower frequency on the order of 0.1 Hz, which modulates the amplitudes of the higher frequencies of vibrations.

In the second simulation, the load was a cyclic disturbance of 100 lb and 100 ft-lb. along each axis of the global frame. The frequency of the load was 3.18 Hz. The simulation was performed once for reference position #1, for which all joint displacements are zero, and once for reference position #2, for which all joints are zero, except for joint 3, which was set at $\pi/2$ radians. The base was allowed to translate. The position of the end effector is shown in Figures 3.1-9a and

3.1-9b, for reference position #1. The magnitudes of the first and second modes for each flexible link are shown in Figures 3.1-10a through 3.1-13b, for reference position #1. The magnitudes of the first and second modes for each flexible link are shown in Figures 3.1-14a through 3.1-17b, for reference position #2.

The third set of simulations were performed for an oscillating load of 1000 lb. and 1000 ft-lb., at a frequency of .318 Hz. The results for the magnitudes of the first and second modes for each flexible link are shown in Figures 3.1-18a through 3.1-18b, for reference position #1. The results for the magnitudes of the first and second modes for each flexible link are shown in Figures 3.1-22a through 3.1-25b, for reference position #1.

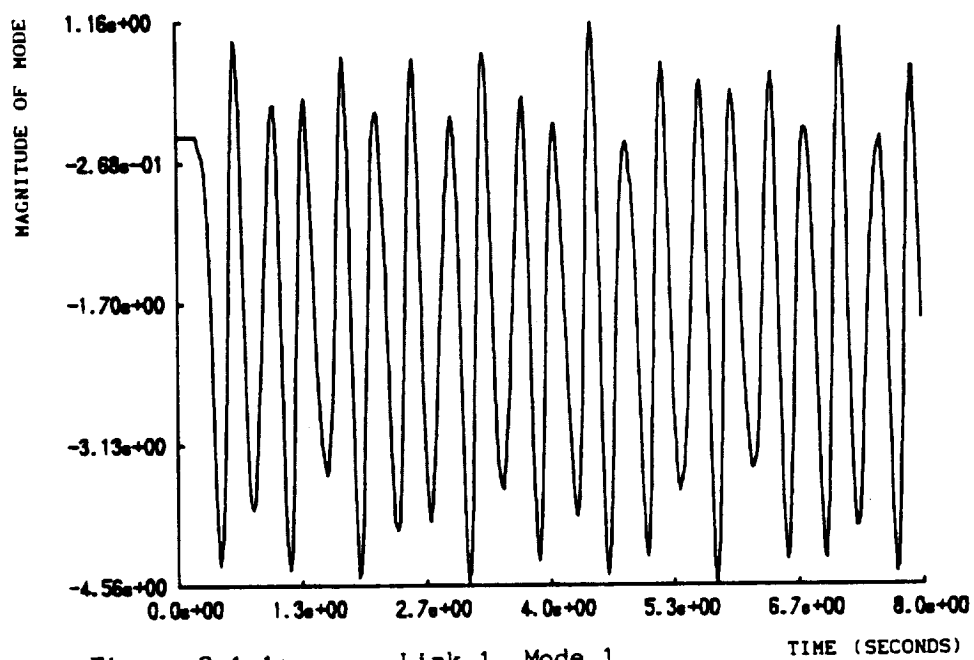


Figure 3.1-1a Link 1, Mode 1

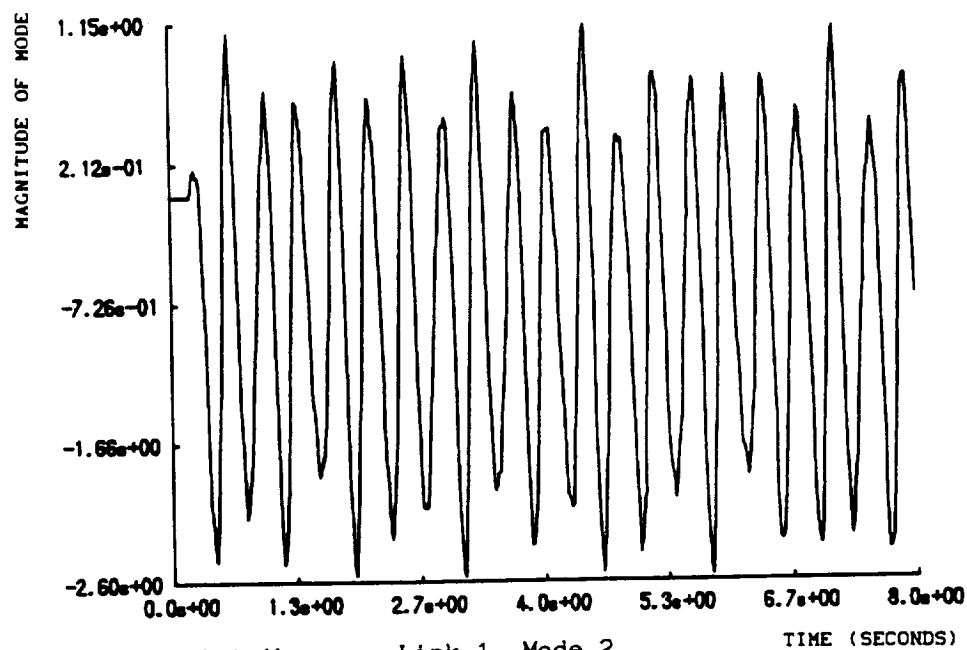


Figure 3.1-1b Link 1, Mode 2

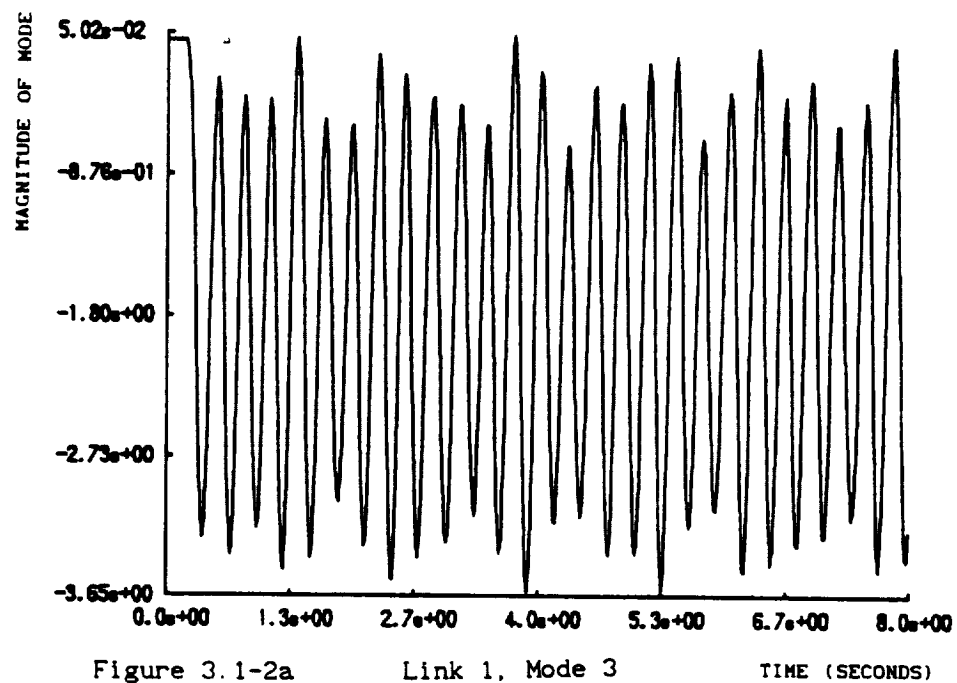


Figure 3.1-2a

Link 1, Mode 3

TIME (SECONDS)

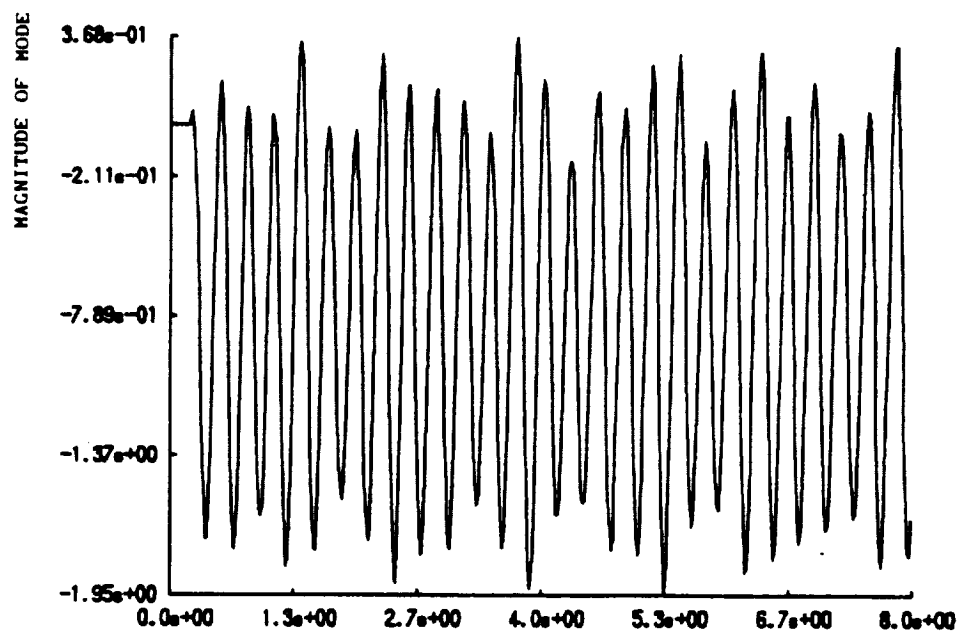


Figure 3.1-2b

Link 1, Mode 4

TIME (SECONDS)

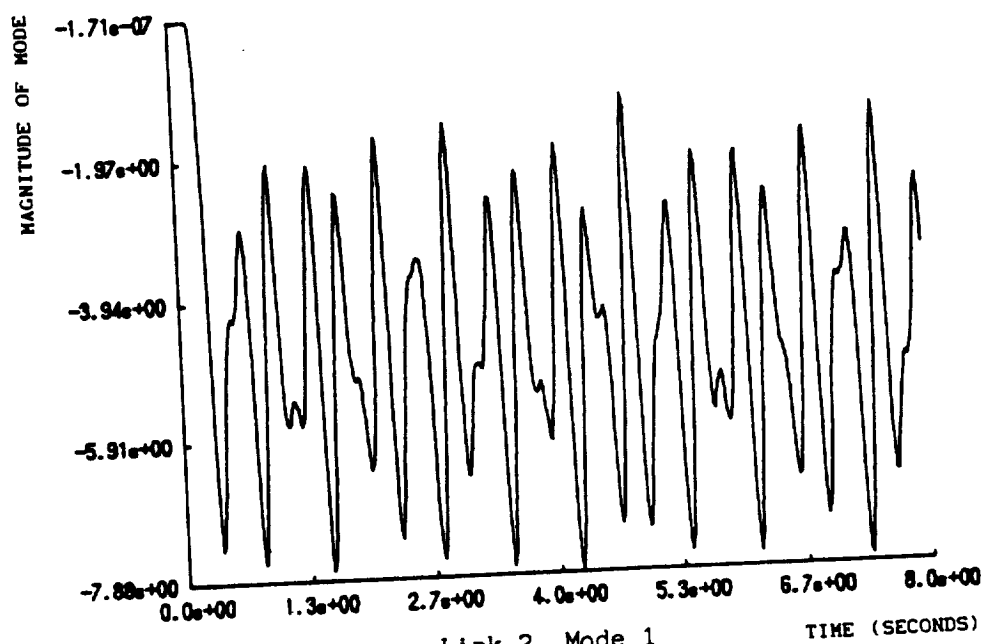


Figure 3.1-3a Link 2, Mode 1

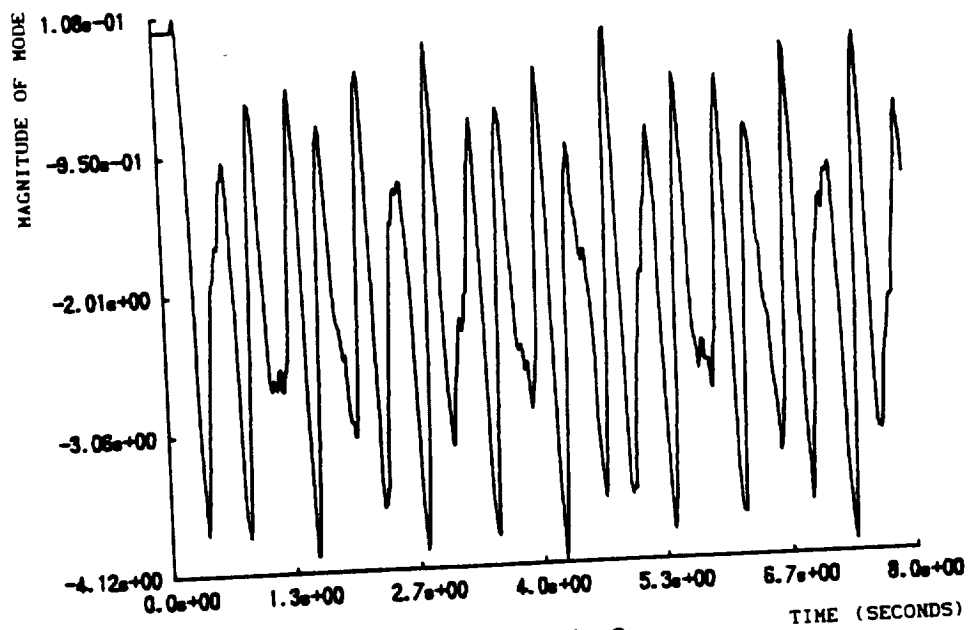
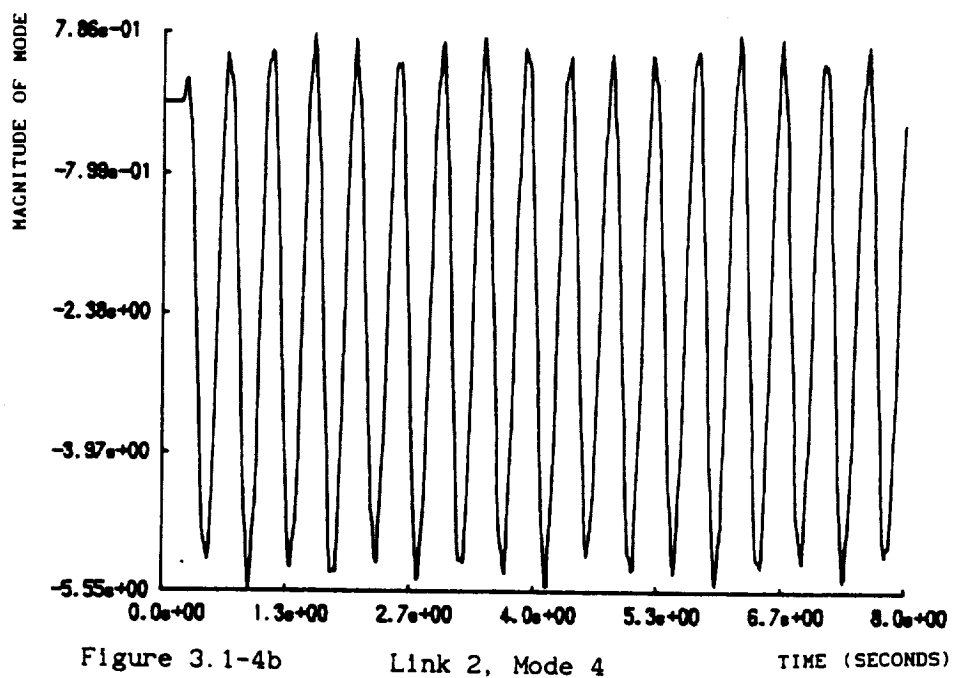
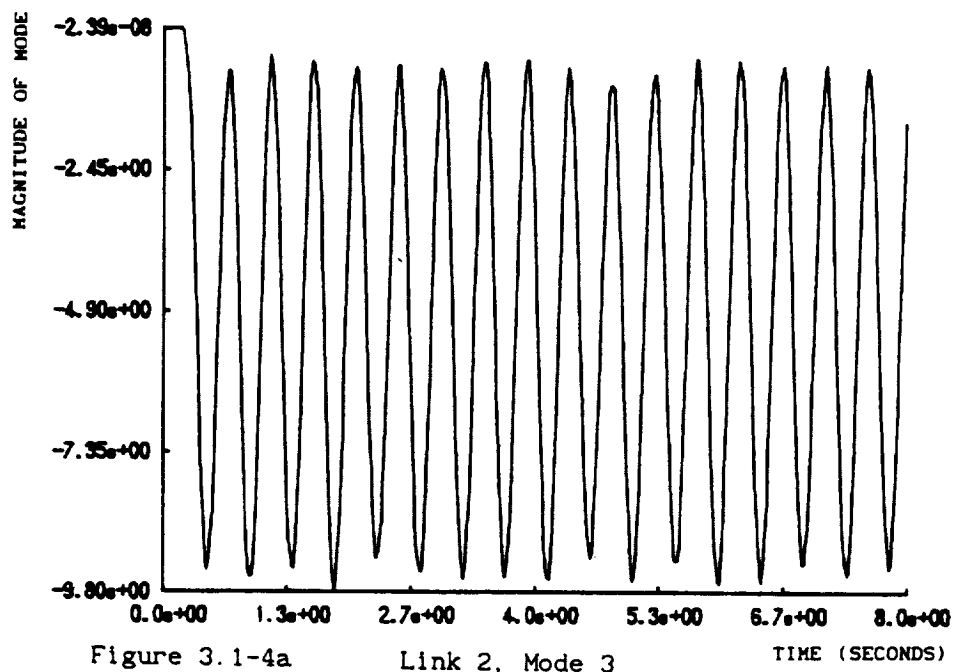


Figure 3.1-3b Link 2, Mode 2



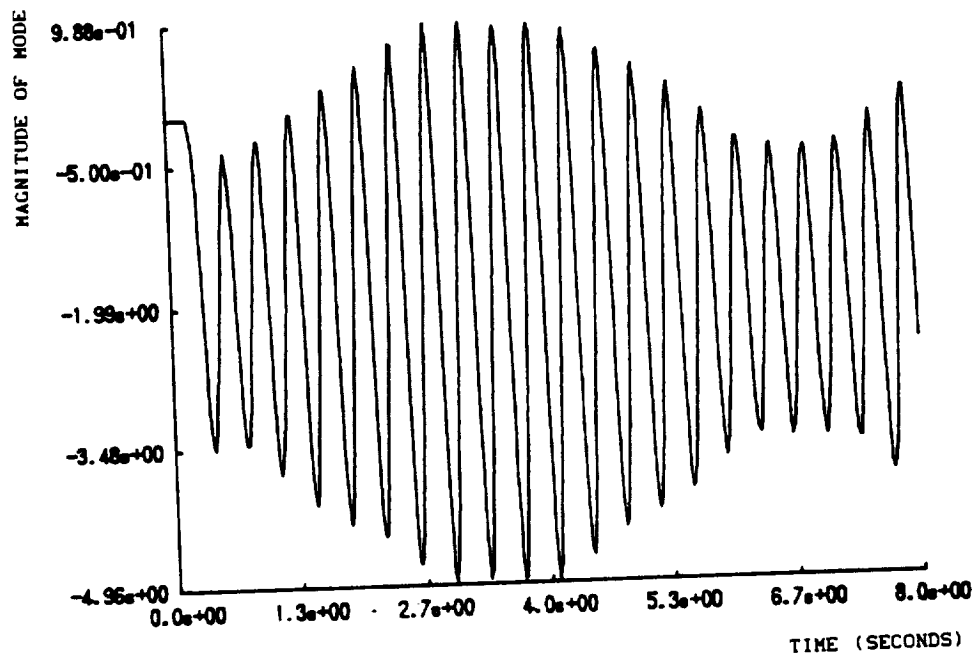


Figure 3.1-5a Link 1, Mode 1

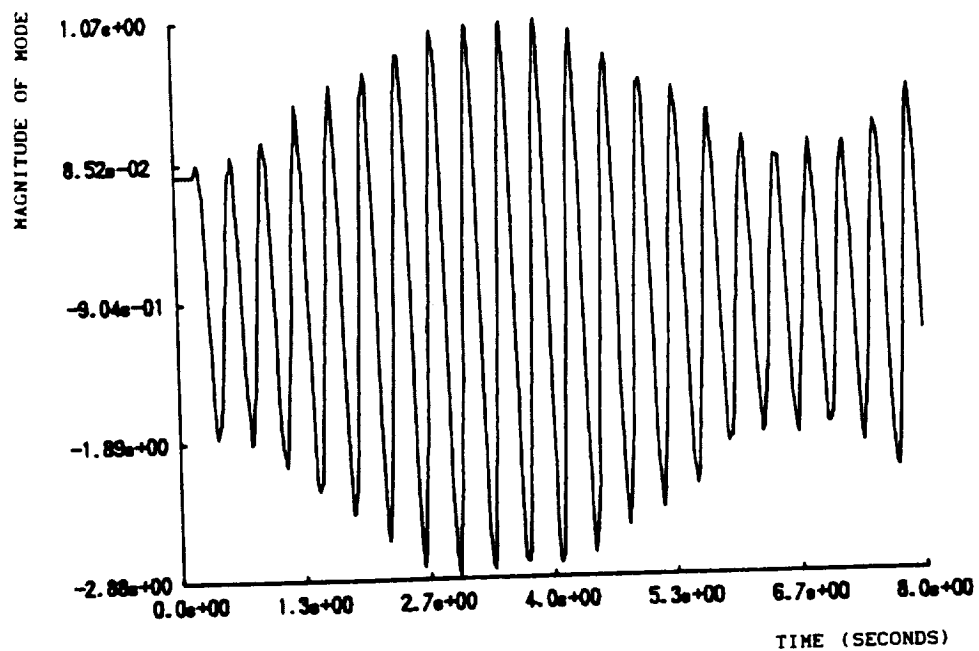
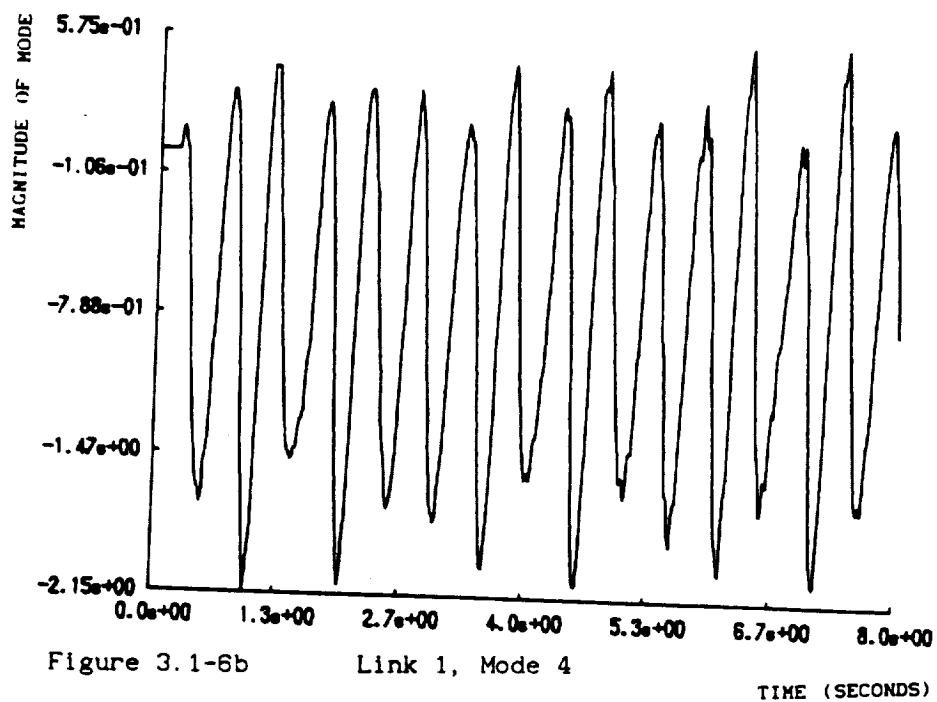
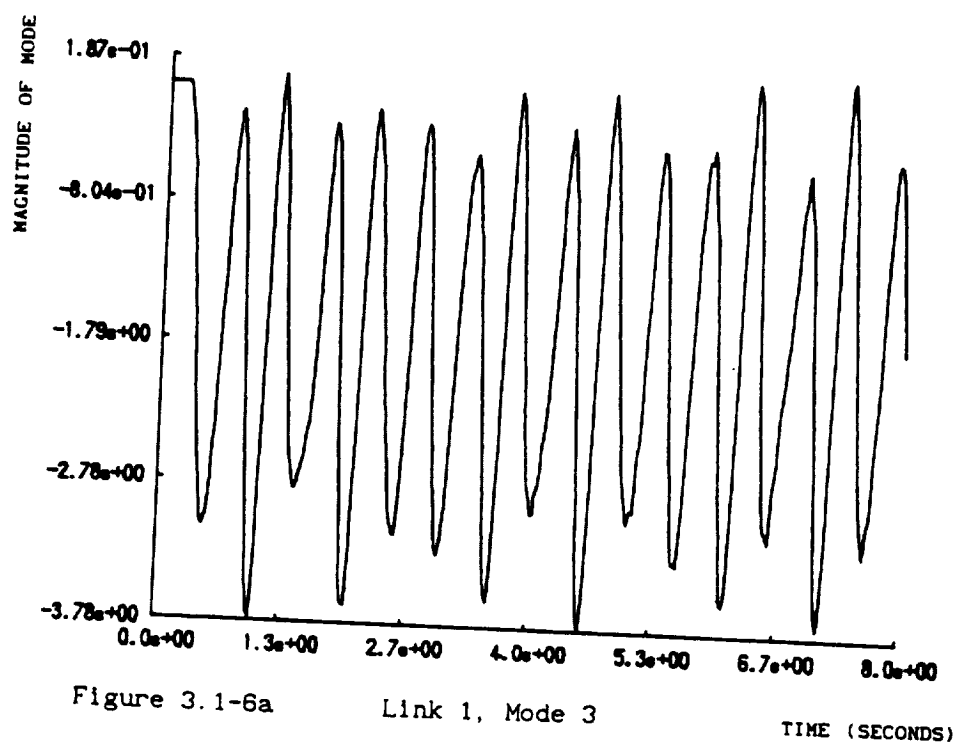
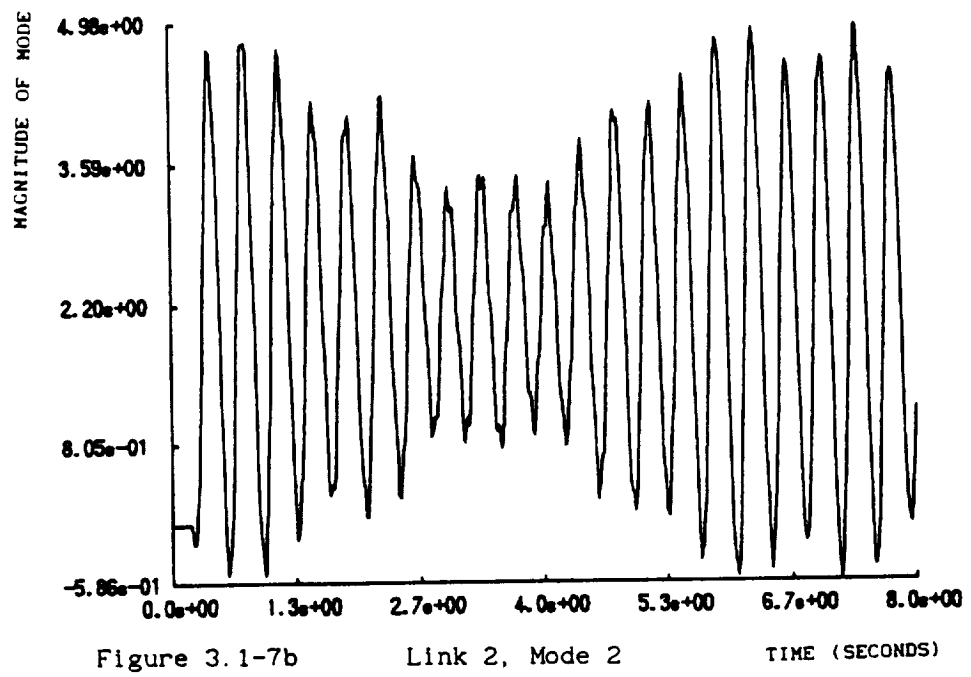
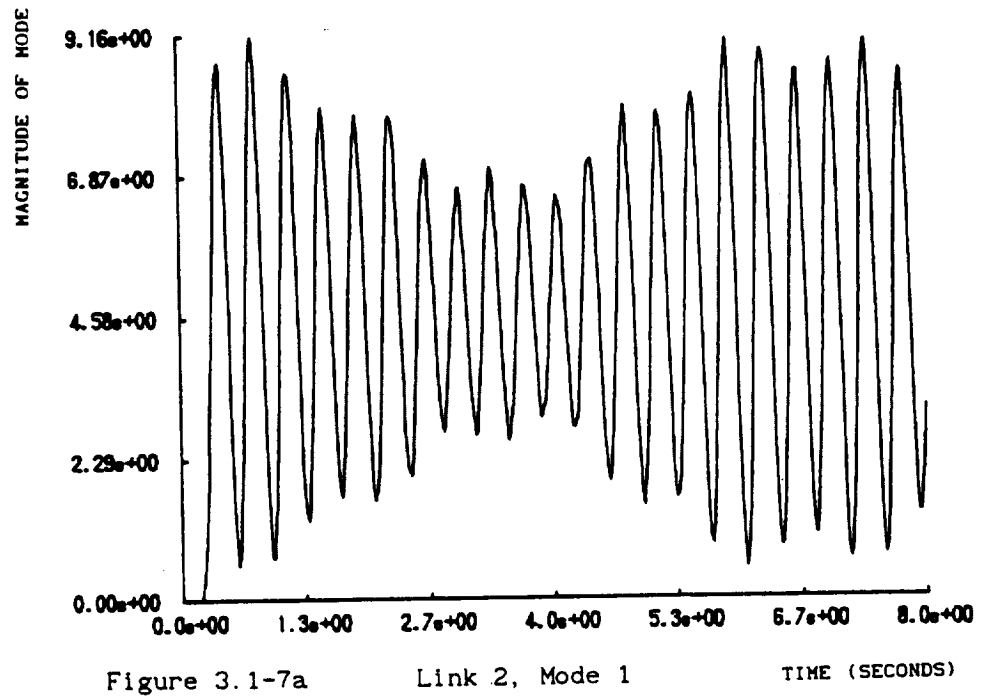


Figure 3.1-5b Link 1, Mode 2





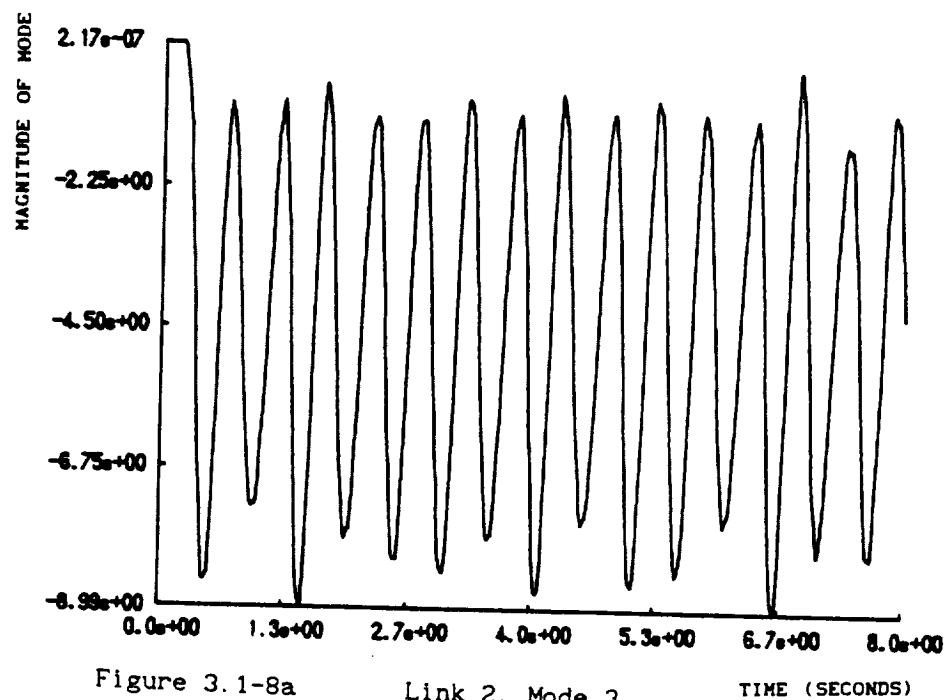


Figure 3.1-8a

Link 2, Mode 3

TIME (SECONDS)

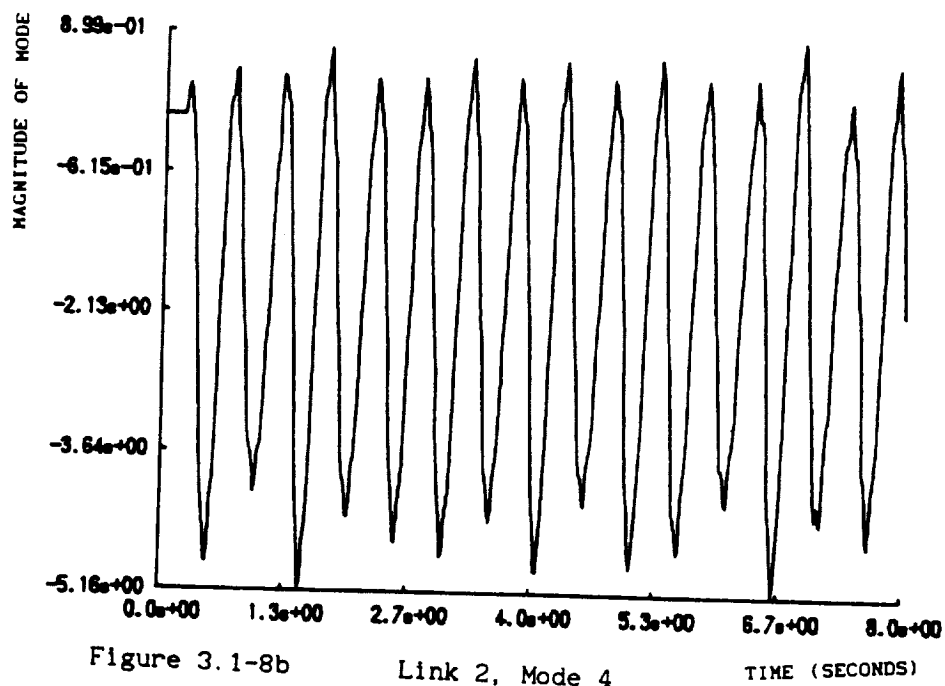


Figure 3.1-8b

Link 2, Mode 4

TIME (SECONDS)

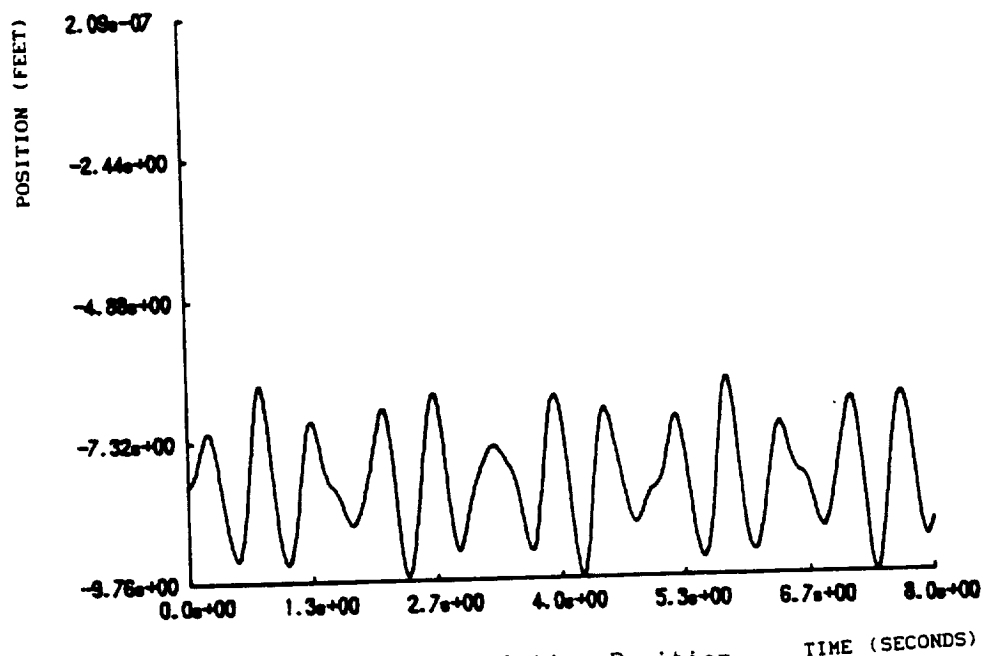


Figure 3.1-9a

Point of Resolution Position

TIME (SECONDS)

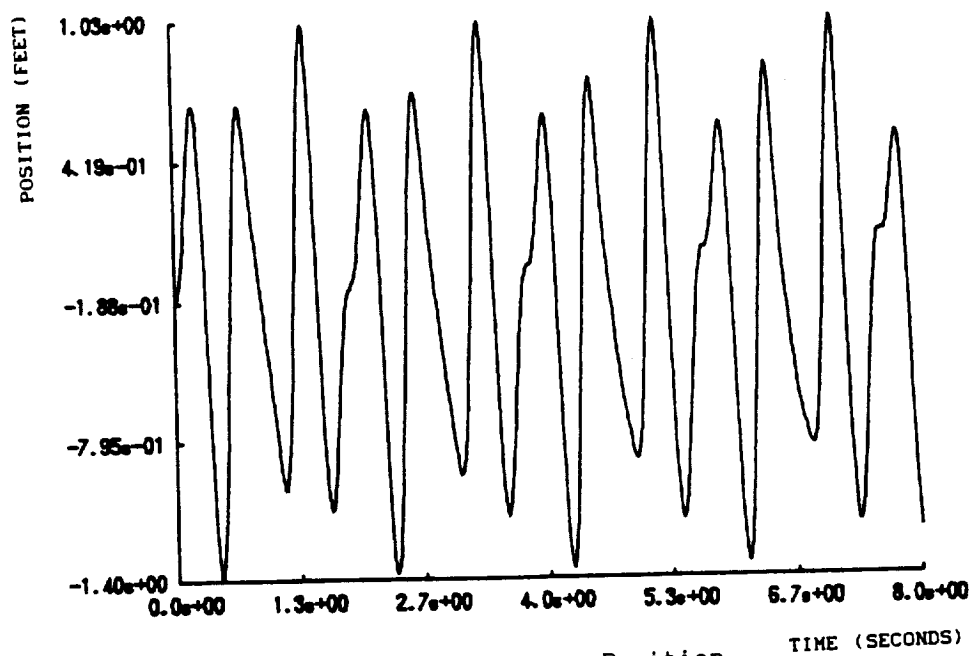
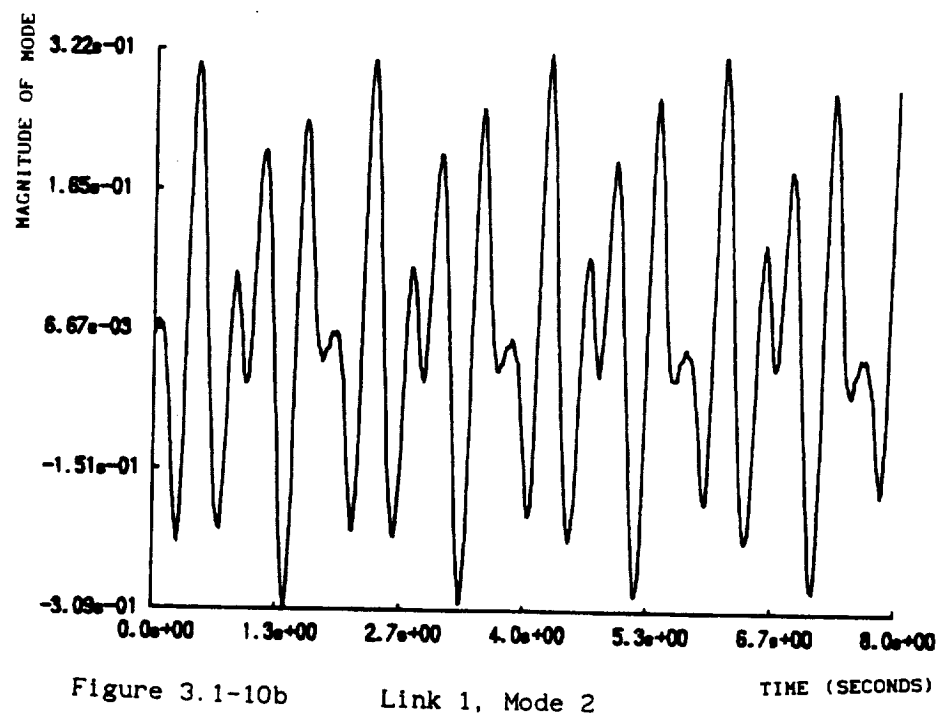
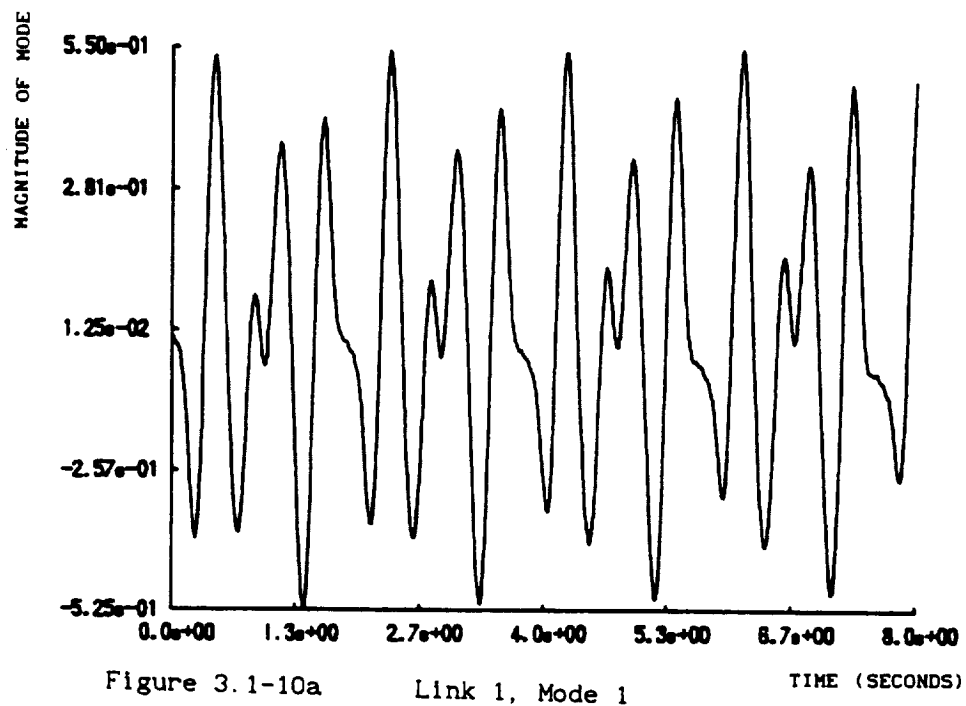


Figure 3.1-9b

Point of Resolution Position

TIME (SECONDS)



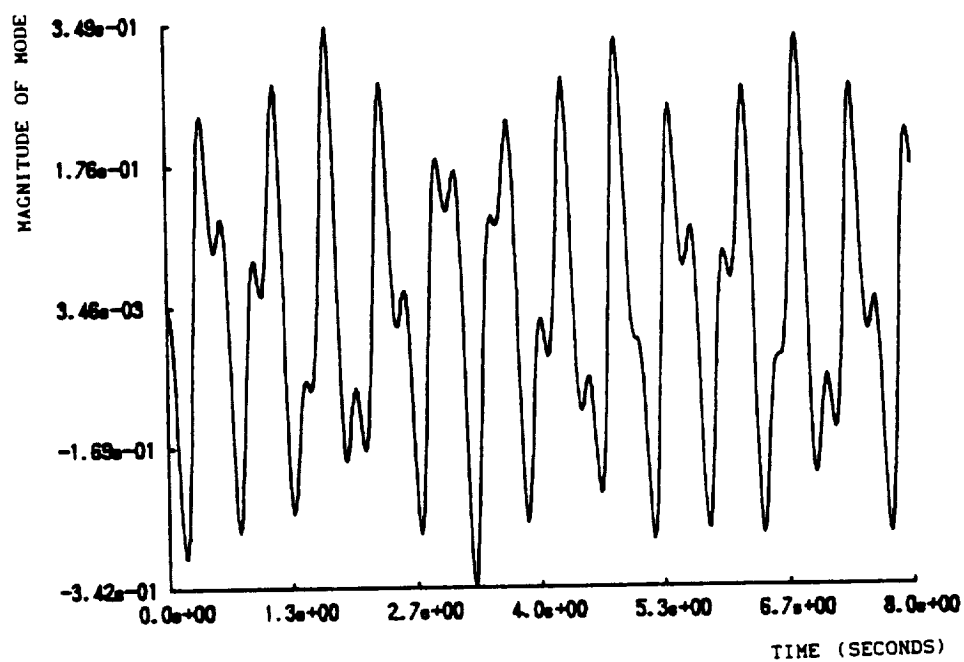


Figure 3.1-11a Link 1, Mode 3

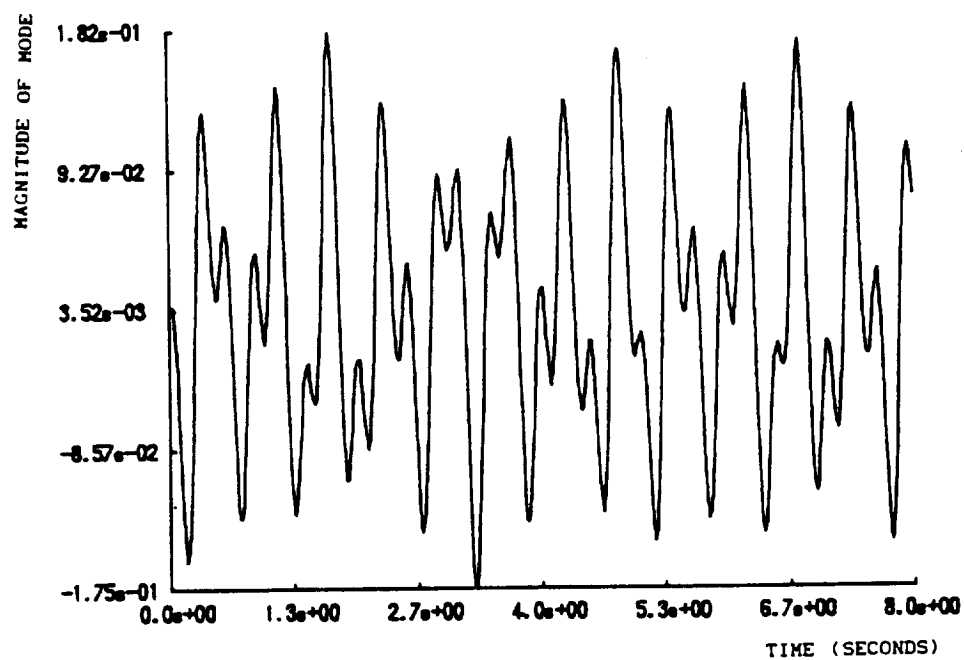


Figure 3.1-11b Link 1, Mode 4

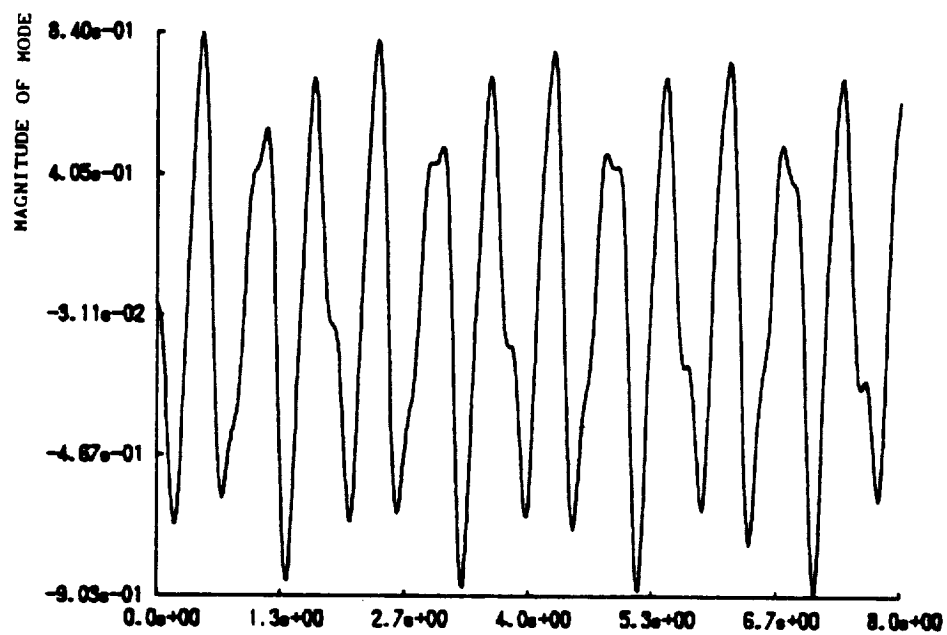


Figure 3.1-12a

Link 2, Mode 1

TIME (SECONDS)

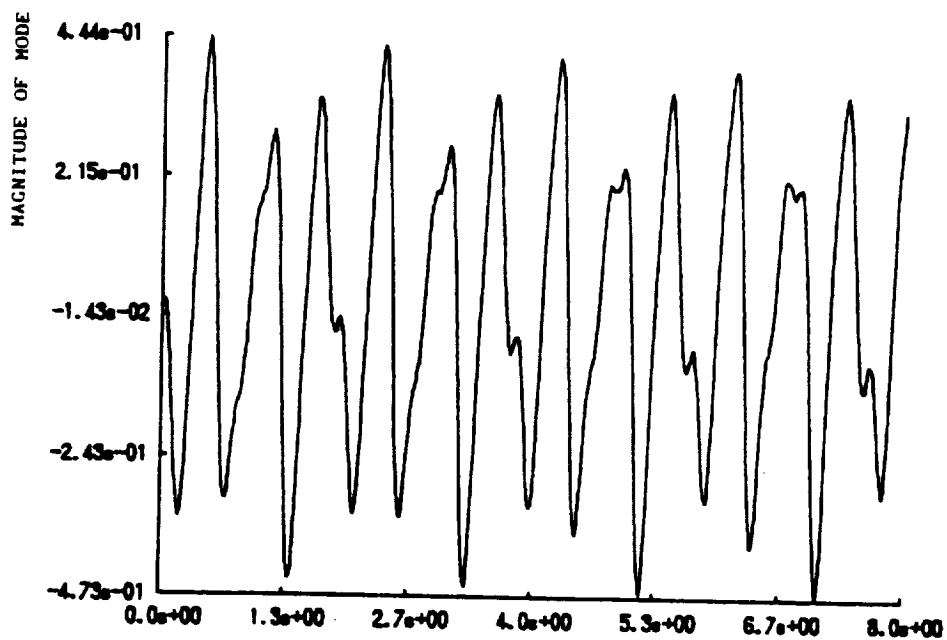
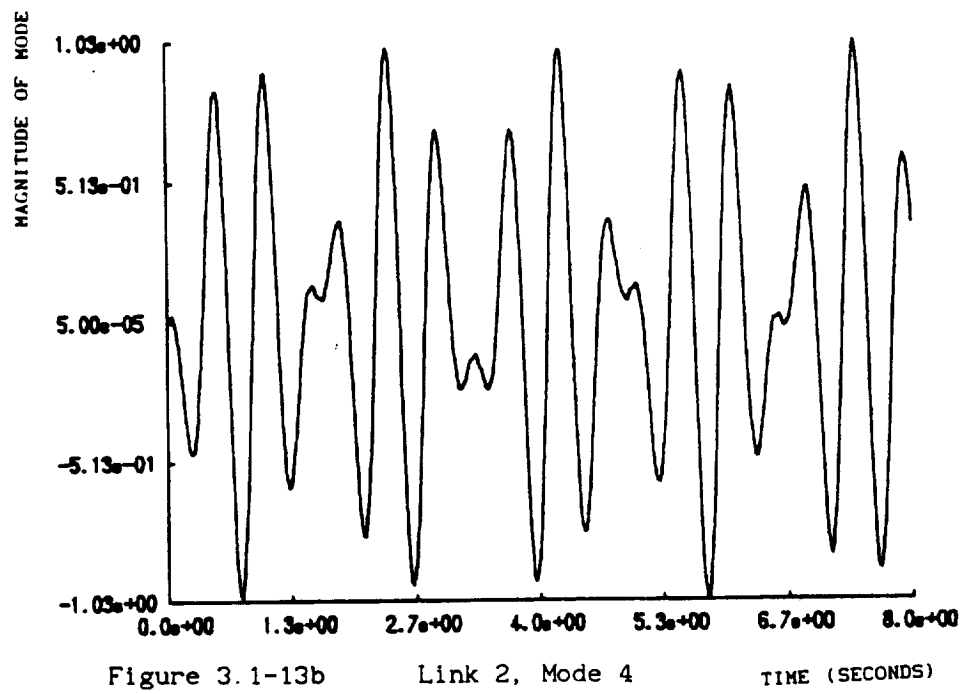
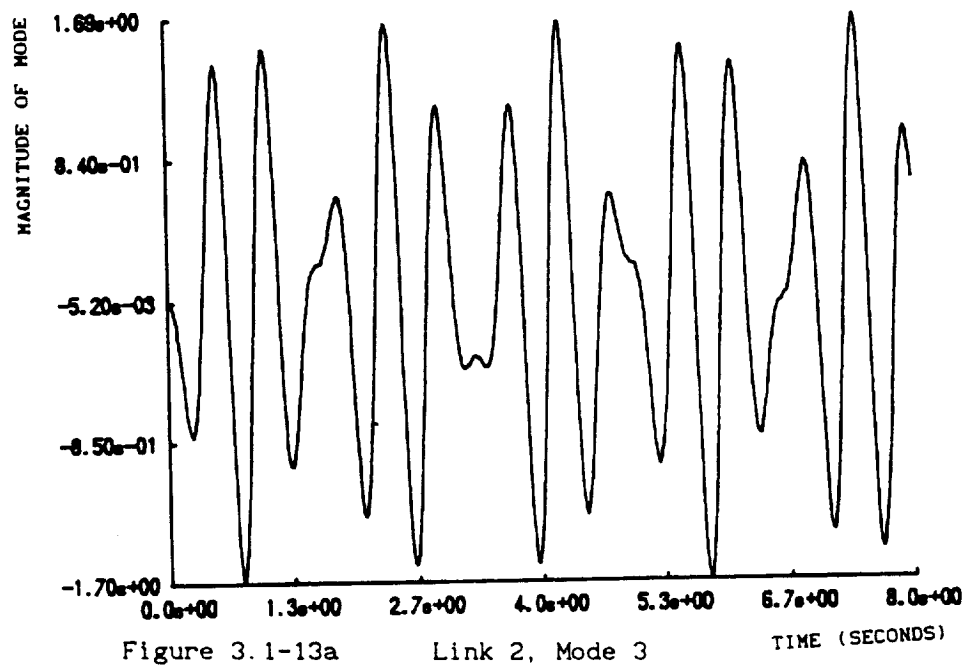
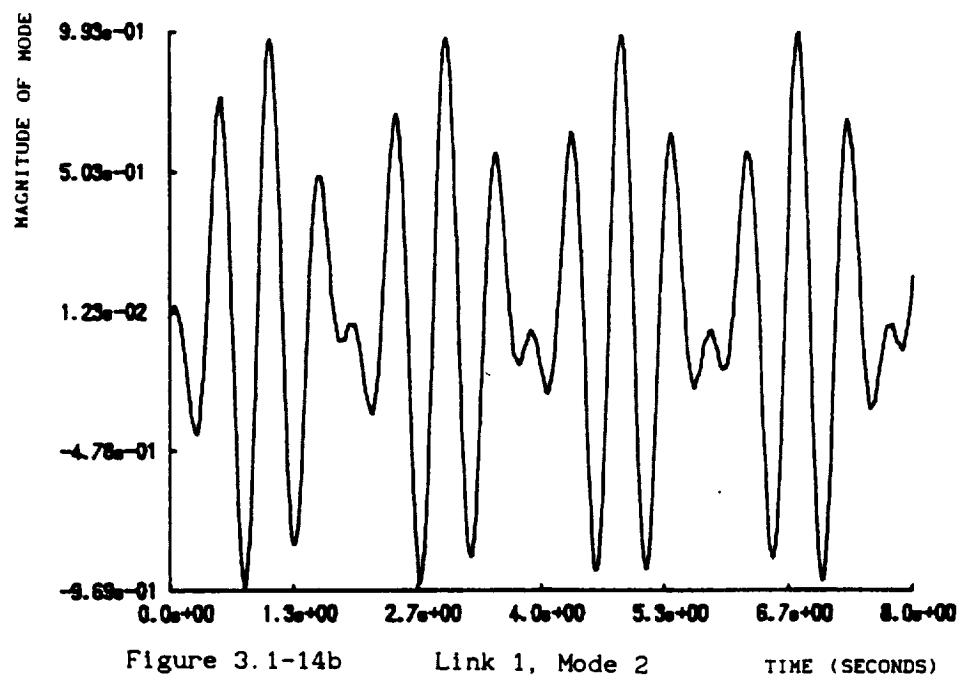
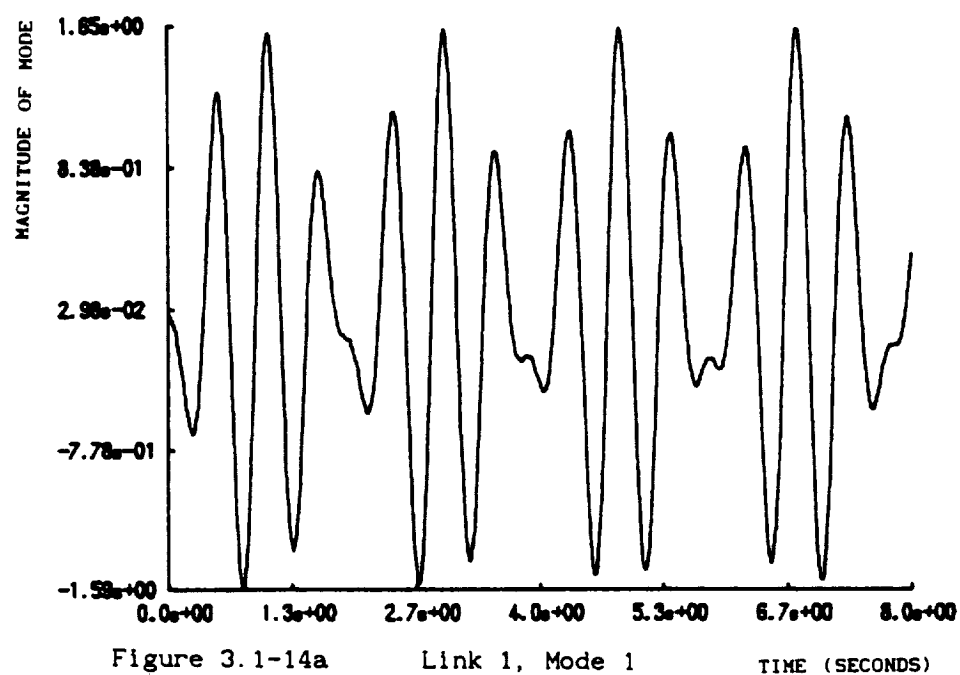


Figure 3.1-12b

Link 2, Mode 2

TIME (SECONDS)





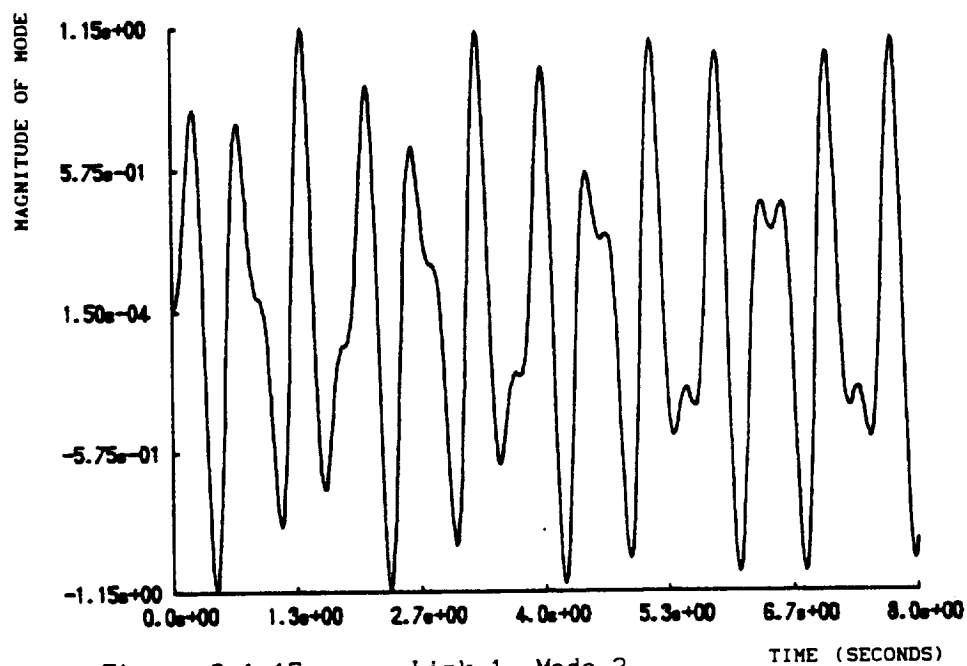


Figure 3.1-15a Link 1, Mode 3

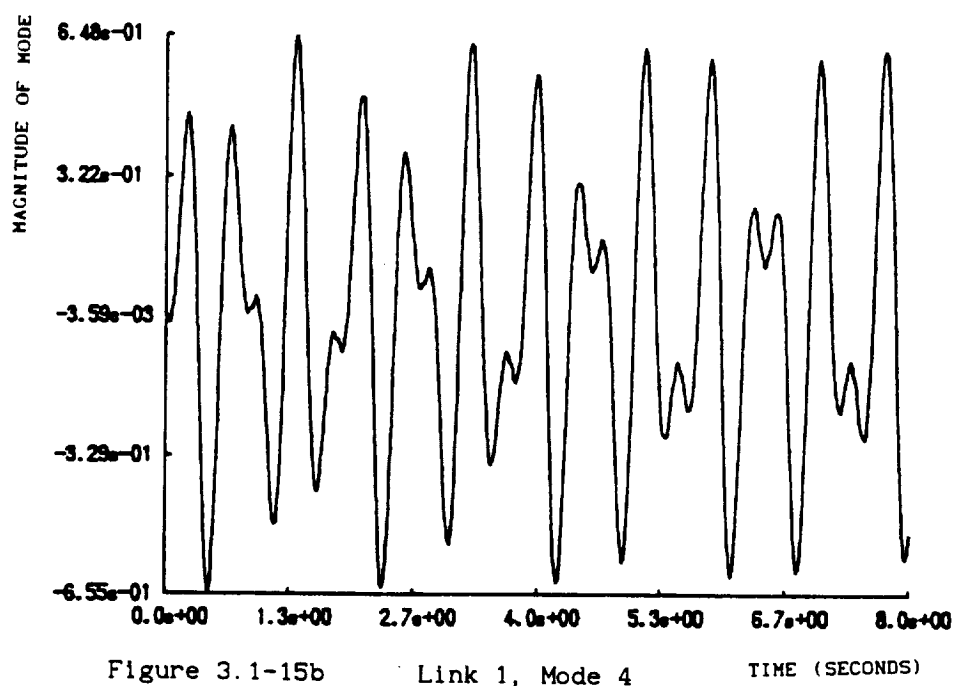
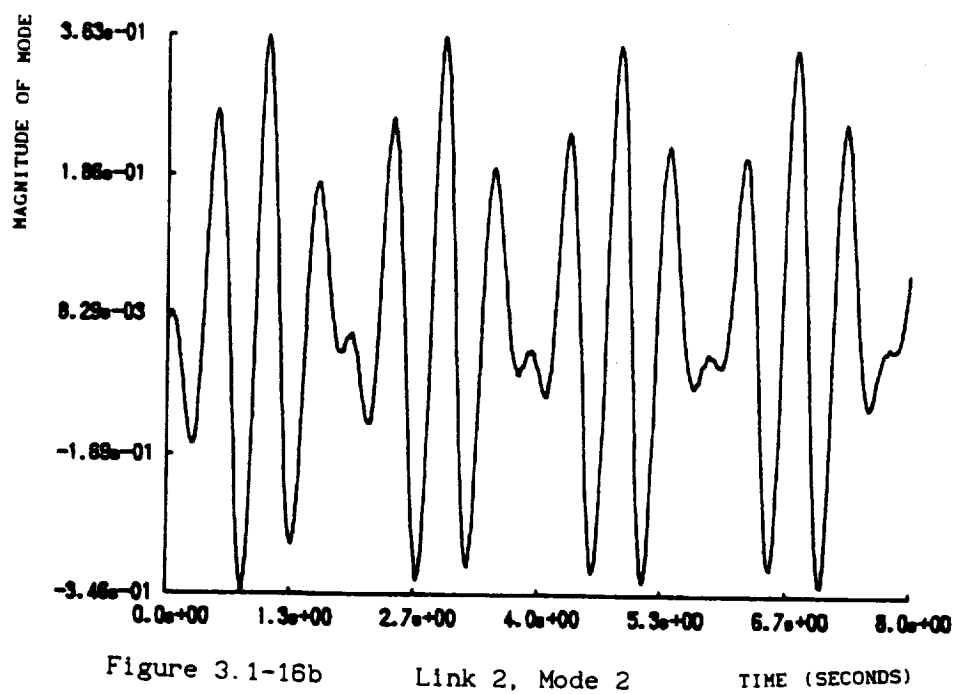
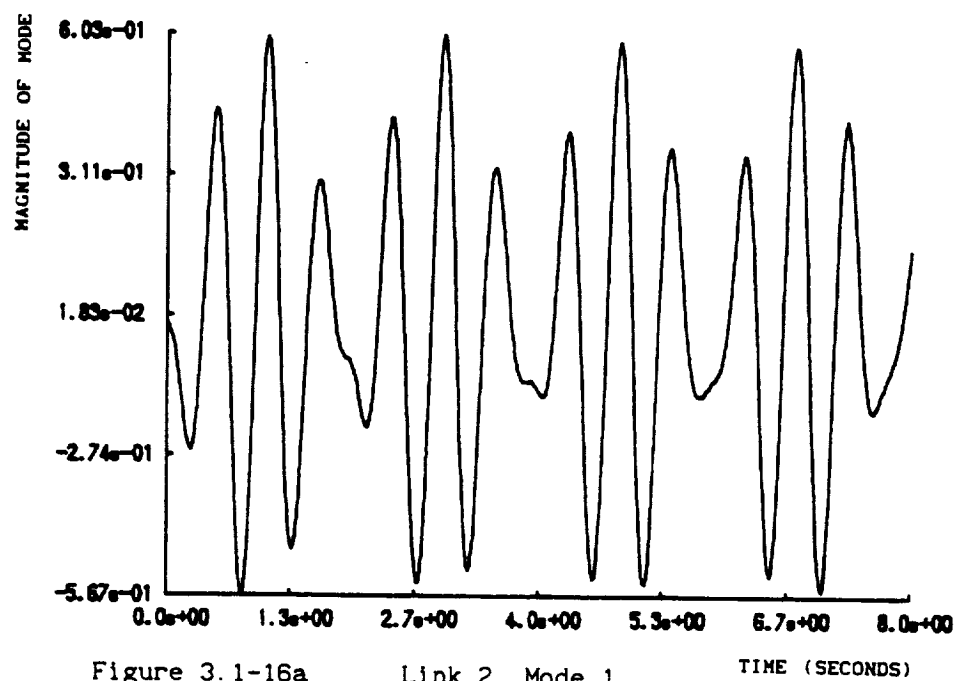
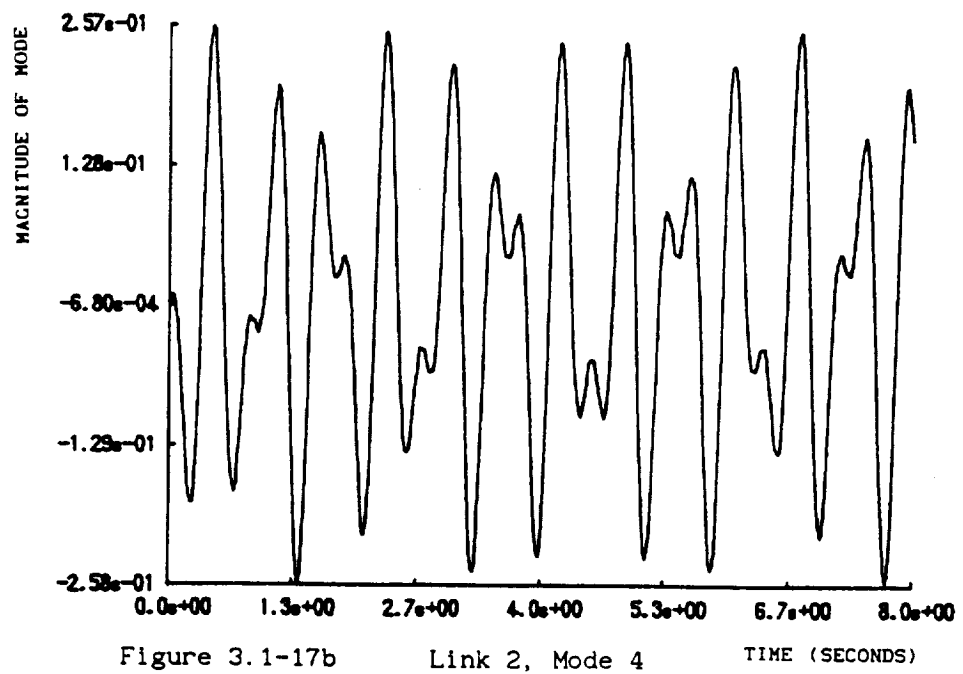
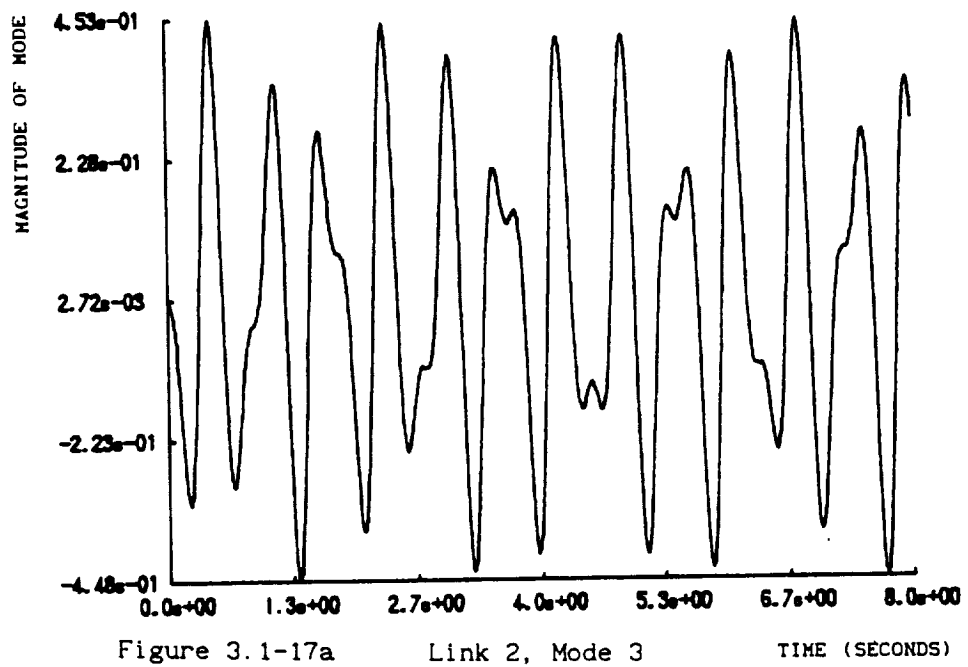
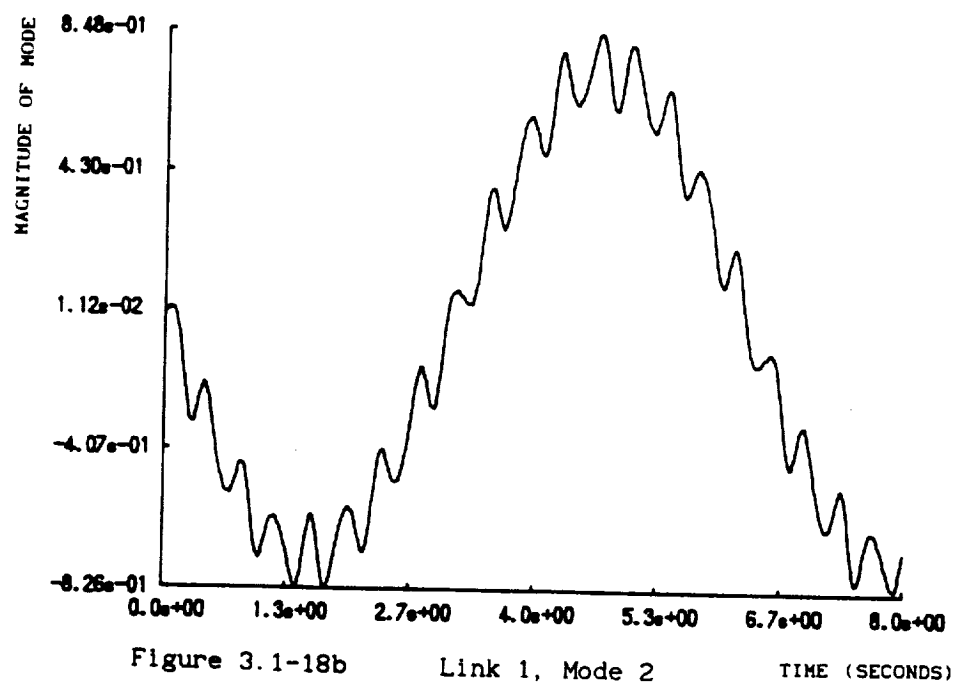
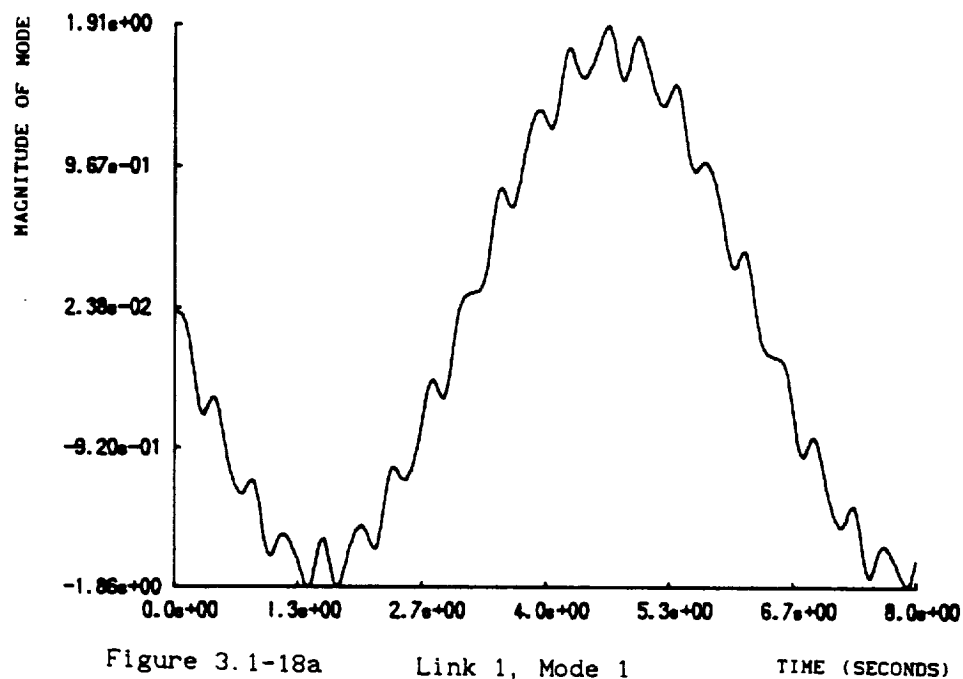


Figure 3.1-15b Link 1, Mode 4







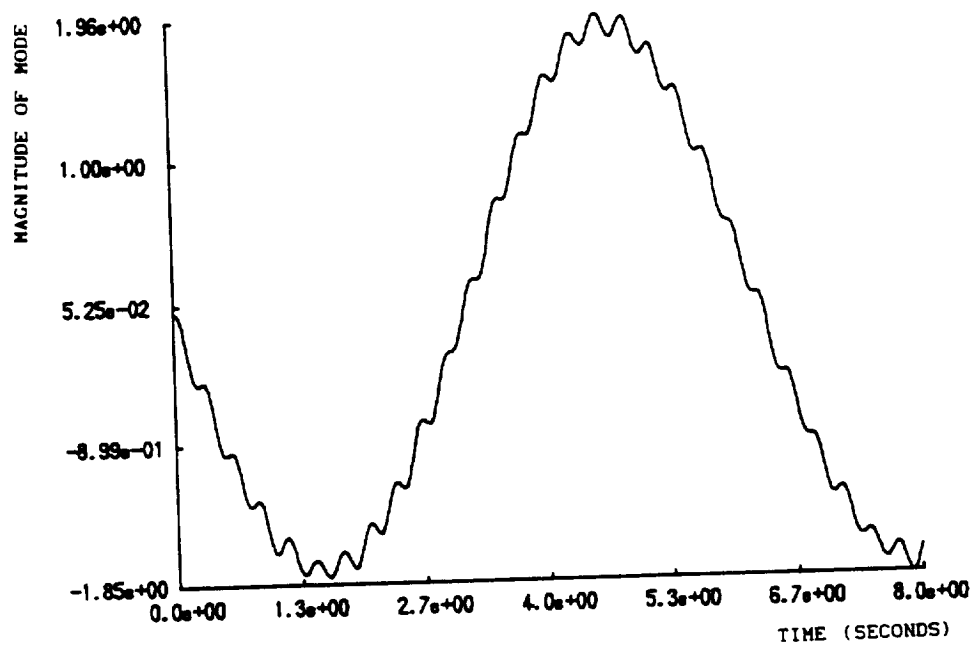


Figure 3.1-19a Link 1, Mode 3

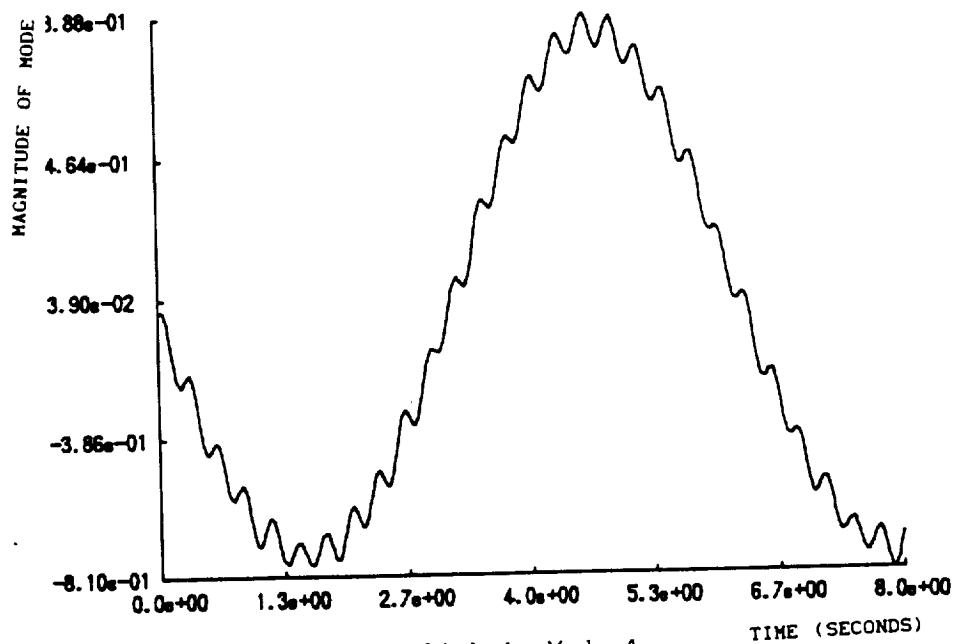
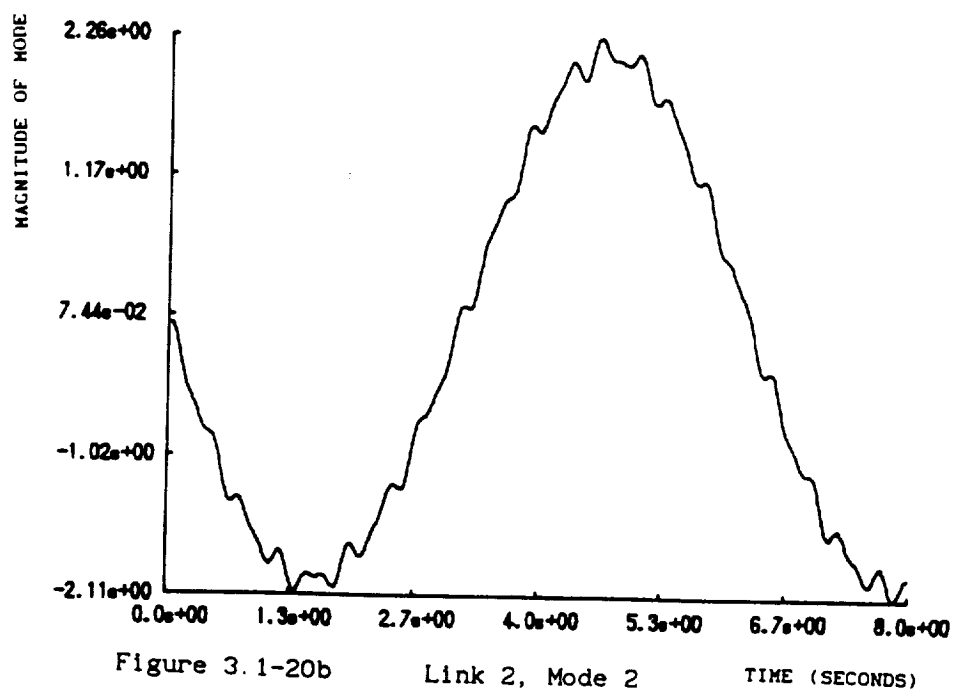
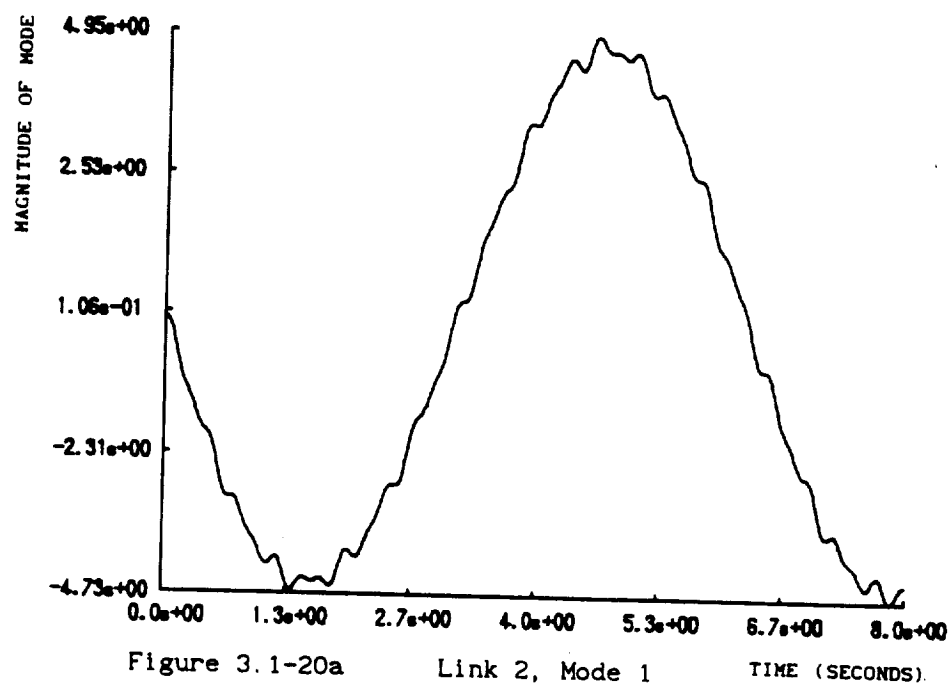
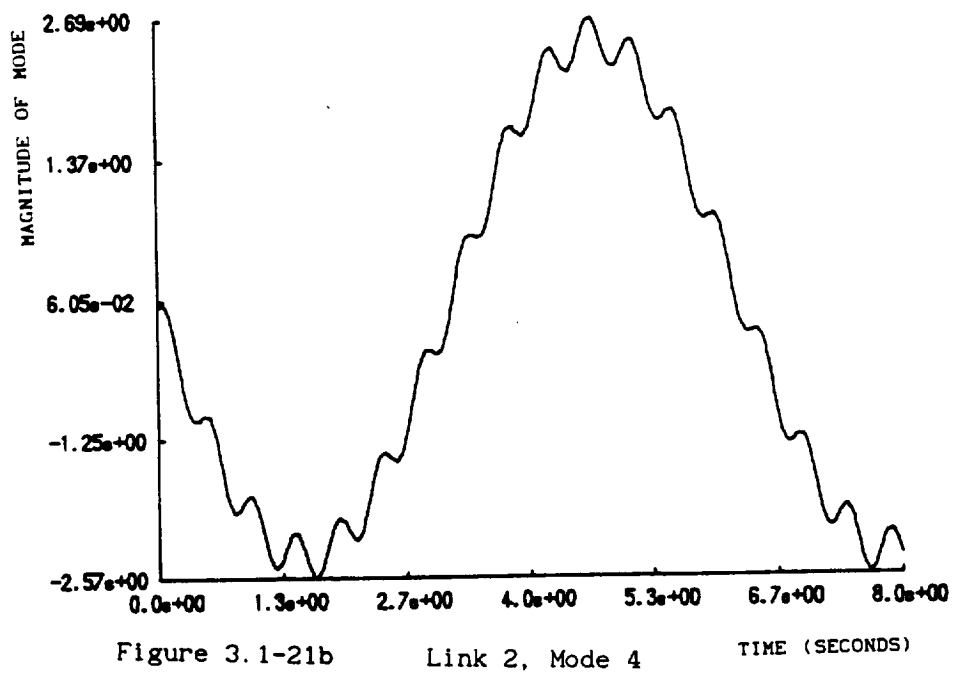
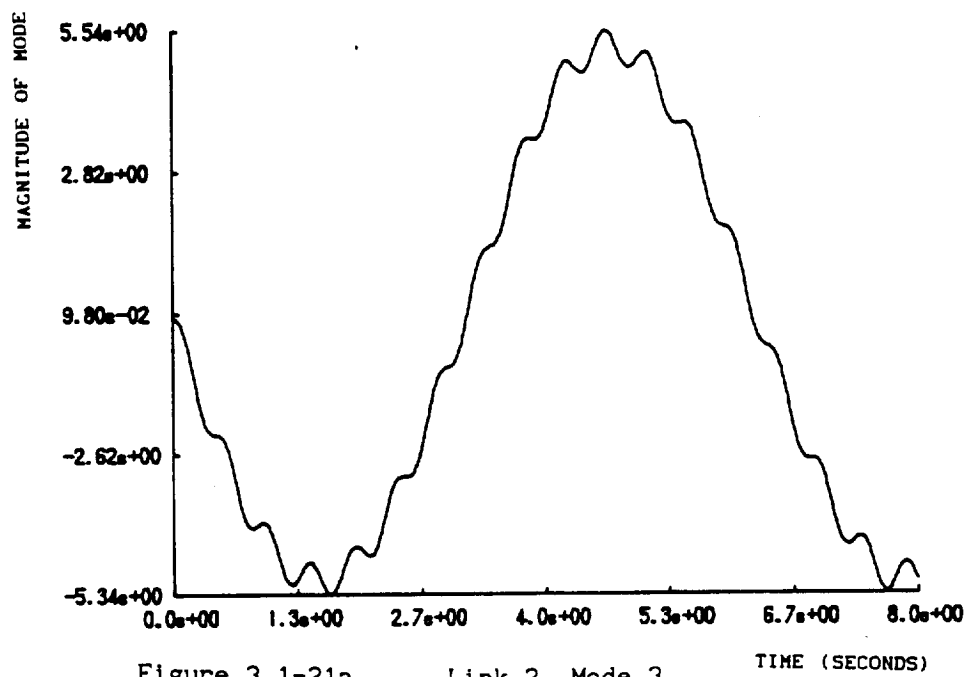
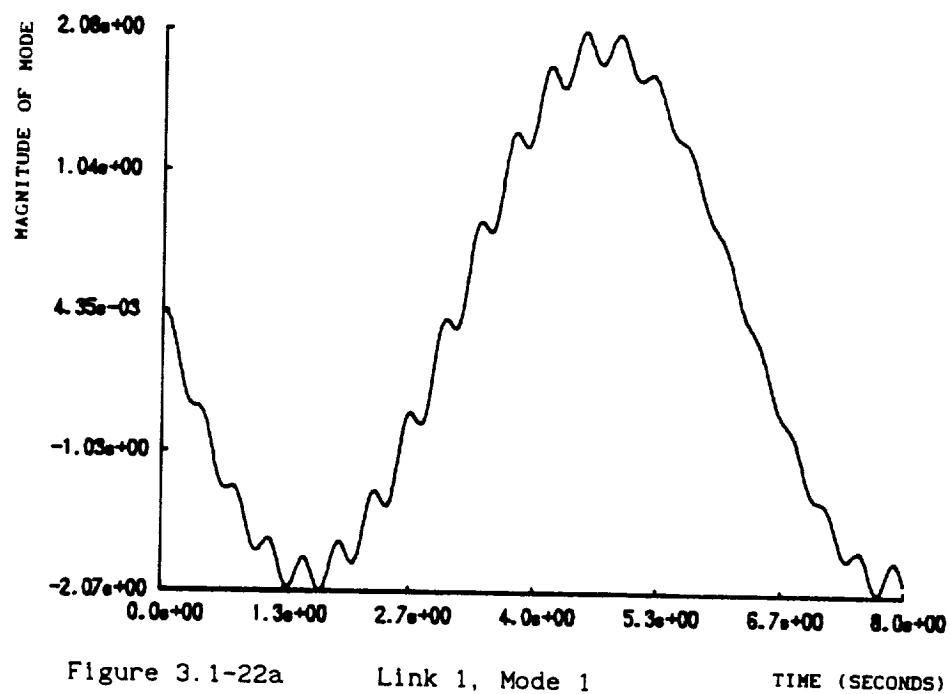


Figure 3.1-19b Link 1, Mode 4







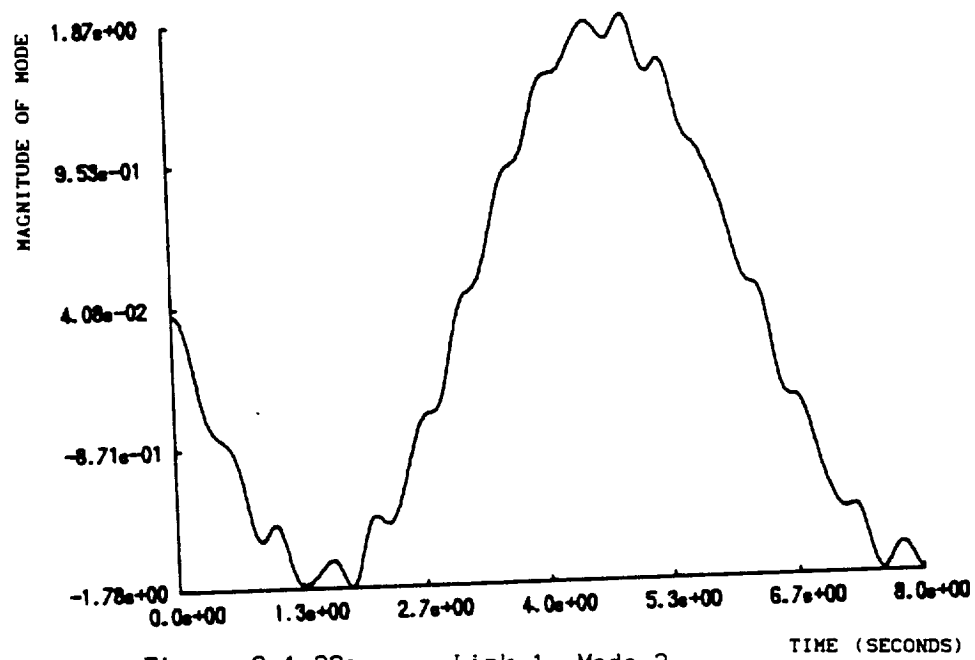


Figure 3.1-23a Link 1, Mode 3

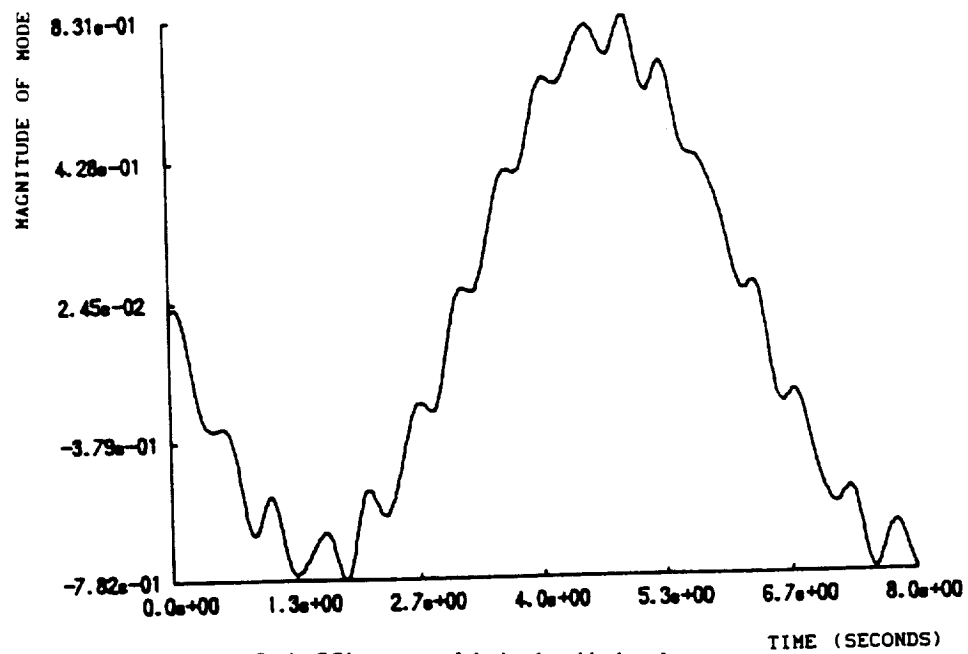
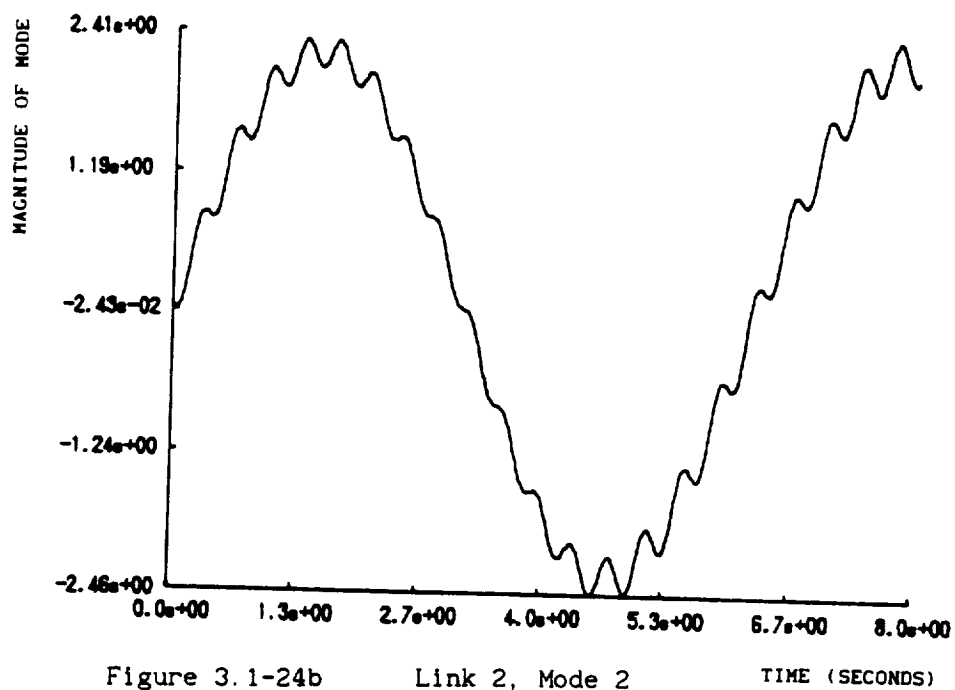
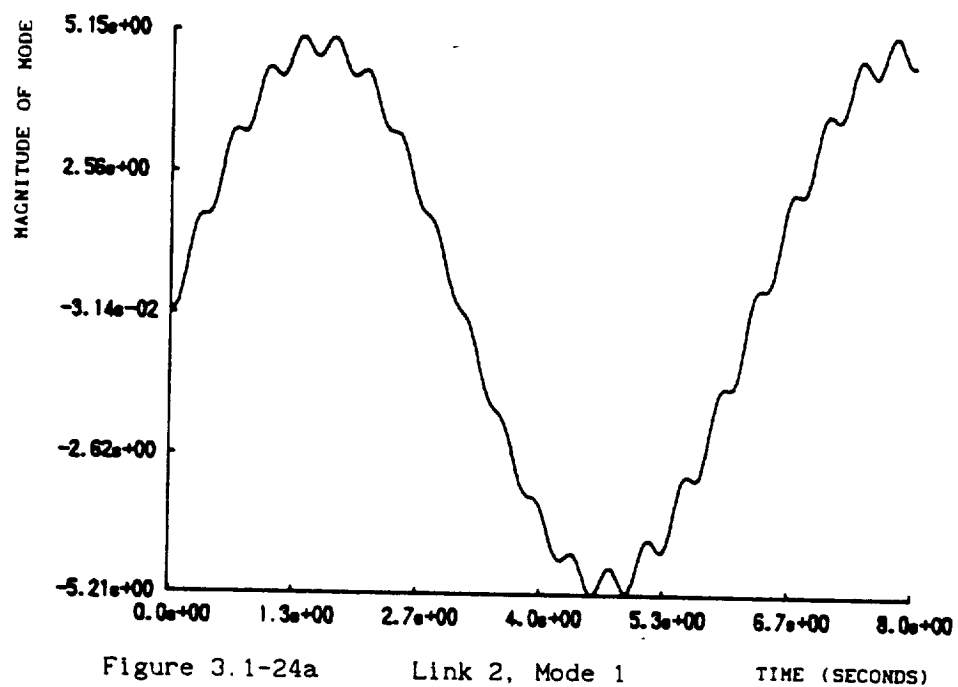


Figure 3.1-23b Link 1, Mode 4



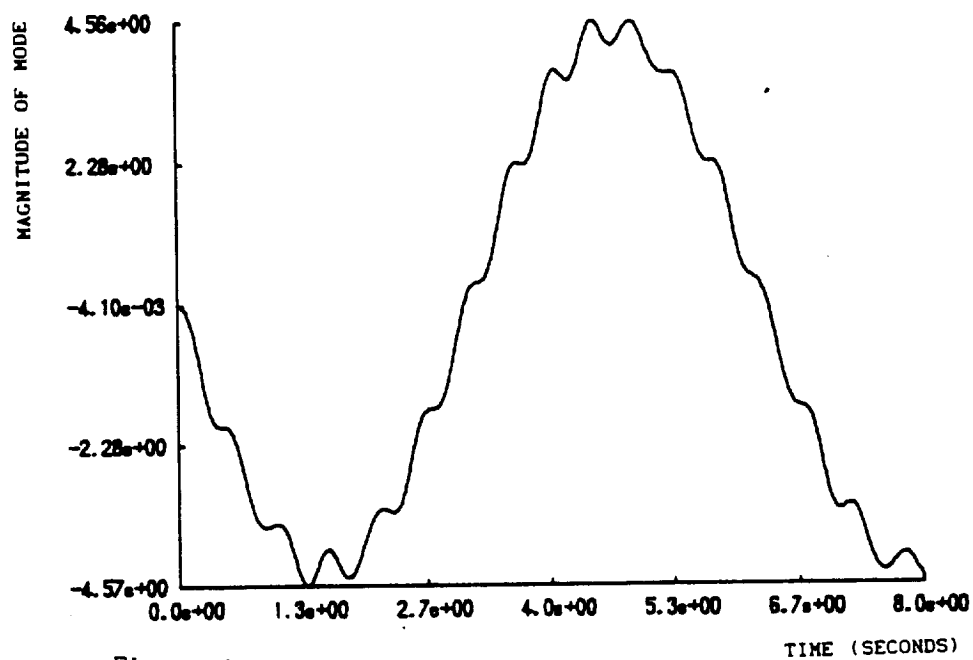


Figure 3.1-25a Link 2, Mode 3

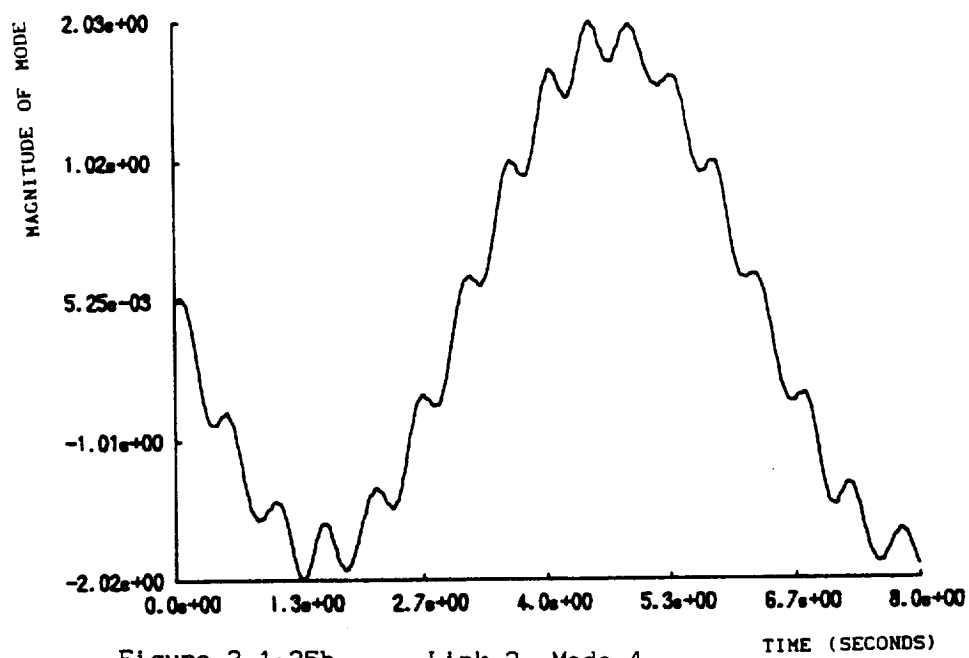


Figure 3.1-25b Link 2, Mode 4

4.0 Conclusions

The flexibility of a robotic manipulator may cause large errors at the end-effector of the system. Dynamic simulation of these vibrations and deflections is necessary to study the control of the system so these vibrations may be eliminated, or reduced. A general modeling method has been presented which includes the dynamics due to gross motions of the base, motions of the joints, and vibrations of the joints and links of the system. The motions of the joints and the base are coupled to the vibrations of the joints and links. The vibrations may be modeled using lumped parameters, truncated mode summation, or a component mode synthesis method.

This model has been implemented in a simulation package called VSim. The package was used to simulate a large space-based manipulator system. Both mode summation and lumped parameter techniques are used in the model. The response of the system was seen to be configuration dependent. Very slow frequencies were present in some configurations, and not in other configurations. The response of the system also was dependent on the type and frequency content of the disturbance applied to the system. It should be noted that the second mode seemed to respond much the same as the lowest mode. This indicates that the system response probably can be modeled

sufficiently by only the first mode.

This simulation package can be easily modified to include a control algorithm. Disturbances may be applied externally, or may be created due to the motion of the links. The number of flexibilities, links, and joints which the simulation package may model is restricted only by the constraints imposed by the computational resource which is used to run the simulation.

APPENDIX A

Derivation of the Local Inertia Matrix

The expression for the kinetic energy of the i th link or sub-link is ideally expressed in integral form as

$$KE_i = \frac{1}{2} \int_V \rho(\underline{x}_i) \left\{ \dot{\underline{R}}(\underline{x}_i) \right\} \cdot \left\{ \dot{\underline{R}}(\underline{x}_i) \right\} dV . \quad (A-1)$$

The velocity, $\dot{\underline{R}}(\underline{x}_i)$, can be written as the sum of the velocity of a reference point and the velocity relative to the reference point. The reference point is chosen to coincide with the local origin of the link. The equation of velocity is then

$$\begin{aligned} {}^{(1)}\dot{\underline{R}}(\underline{x}_i) &= {}^{(1)}\dot{\underline{R}}(\underline{Q}_1) + {}^{(1)}\dot{\underline{d}}(\underline{x}_i) + {}^{(1)}\underline{\omega}(\underline{Q}_1) \times \left\{ {}^{(1)}\underline{d}(\underline{x}_i) + {}^{(1)}\underline{x}_i \right\} \\ &= {}^{(1)}\dot{\underline{R}}(\underline{Q}_1) + {}^{(1)}\underline{\omega}(\underline{Q}_1) \times \left\{ {}^{(1)}\underline{x}_i \right\} \\ &\quad + {}^{(1)}\underline{\omega}(\underline{Q}_1) \times \left\{ {}^{(1)}\underline{d}(\underline{x}_i) \right\} + {}^{(1)}\dot{\underline{d}}(\underline{x}_i) \end{aligned} \quad (A-2)$$

where ${}^{(1)}\dot{\underline{R}}(\underline{Q}_1)$ is the translational velocity of the local coordinate system, ${}^{(1)}\underline{\omega}(\underline{Q}_1)$ is the rotational velocity of the local coordinate system, ${}^{(1)}\dot{\underline{d}}(\underline{x}_i)$ is the velocity of the point of interest relative to the local coordinate frame, and ${}^{(1)}\underline{d}(\underline{x}_i) + {}^{(1)}\underline{x}_i$ is the distance from the local coordinate frame to the point of interest. For notational brevity, define

$${}^{(1)}\underline{g}(\underline{x}_i) = {}^{(1)}\underline{x}_i - {}^{(1)}\underline{cm}_1 . \quad (A-3)$$

Also, the pre-script ${}^{(1)}$ will be dropped, noting that all vectors in this appendix will be referenced to the local frame.

Substituting the expression for $\dot{\underline{R}}(\underline{x}_1)$, the kinetic energy can be expressed as:

$$\begin{aligned}
 KE_1 = & \frac{1}{2} \int_V \rho(\underline{x}_1) \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} \cdot \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} dV \\
 & + \int_V \rho(\underline{x}_1) \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} \cdot \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{x}_1 \right\} dV \\
 & + \int_V \rho(\underline{x}_1) \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} \cdot \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{d}(\underline{x}_1) \right\} dV \\
 & + \int_V \rho(\underline{x}_1) \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} \cdot \left\{ \dot{\underline{d}}(\underline{x}_1) \right\} dV \\
 & + \frac{1}{2} \int_V \rho(\underline{x}_1) \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{x}_1 \right\} \cdot \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{x}_1 \right\} dV \\
 & + \int_V \rho(\underline{x}_1) \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{x}_1 \right\} \cdot \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{d}(\underline{x}_1) \right\} dV \\
 & + \int_V \rho(\underline{x}_1) \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{x}_1 \right\} \cdot \left\{ \dot{\underline{d}}(\underline{x}_1) \right\} dV \\
 & + \frac{1}{2} \int_V \rho(\underline{x}_1) \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{d}(\underline{x}_1) \right\} \cdot \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{d}(\underline{x}_1) \right\} dV \\
 & + \int_V \rho(\underline{x}_1) \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{d}(\underline{x}_1) \right\} \cdot \left\{ \dot{\underline{d}}(\underline{x}_1) \right\} dV \\
 & + \frac{1}{2} \int_V \rho(\underline{x}_1) \left\{ \dot{\underline{d}}(\underline{x}_1) \right\} \cdot \left\{ \dot{\underline{d}}(\underline{x}_1) \right\} dV
 \end{aligned} \tag{A-4}$$

These integrals define the total kinetic energy of a link including that due to gross motion, vibrations, and the coupling between the gross motion and vibrations. Each integral can be rewritten so that the velocities are factored out of the integral. The integrations are performed off-line to predict the inertia terms that are needed for the dynamic model.

The first kinetic energy term,

$$\begin{aligned}
 & \frac{1}{2} \int_V \rho(\underline{x}_1) \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} \cdot \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} dV \\
 &= \frac{1}{2} \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\}^T \int_V \rho(\underline{x}_1) dV \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} \\
 &= \frac{1}{2} \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\}^T m_1 \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\}, \tag{A-5}
 \end{aligned}$$

letting $\left[I_{mm_1} \right] = m_1 [I]$, this kinetic energy term becomes:

$$= \frac{1}{2} \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\}^T \left[I_{mm} \right] \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} \tag{A-6}$$

This term is the kinetic energy of the link moving as a point mass.

The second kinetic energy term is

$$\begin{aligned}
 & \int_V \rho(\underline{x}_1) \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} \cdot \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{x}_1 \right\} dV + \\
 &= \int_V \rho(\underline{x}_1) \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} \cdot \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{cm}_1 \right\} dV + \\
 & \int_V \rho(\underline{x}_1) \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} \cdot \left\{ \underline{\omega}(\underline{Q}_1) \times \left\{ \underline{x}_1 - \underline{cm}_1 \right\} \right\} dV \\
 &= \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} \cdot \left\{ \underline{\omega}(\underline{Q}_1) \times \left[\int_V \rho(\underline{x}_1) \underline{cm}_1 dV \right] \right\} + \\
 & \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} \cdot \left\{ \underline{\omega}(\underline{Q}_1) \times \int_V \rho(\underline{x}_1) \left\{ \underline{x}_1 - \underline{cm}_1 \right\} dV \right\}
 \end{aligned}
 \tag{A-7}$$

The second integration in this kinetic energy term is zero because

$$\begin{aligned}
 \int_V \rho(\underline{x}_1) \left\{ \underline{x}_1 - \underline{cm}_1 \right\} dV &= \int_V \rho(\underline{x}_1) \underline{x}_1 dV - \int_V \rho(\underline{x}_1) \underline{cm}_1 dV \\
 &= m_1 \underline{cm}_1 - m_1 \underline{cm}_1 = 0 .
 \end{aligned}
 \tag{A-8}$$

The first integration can be written in more convenient notation,

$$\begin{aligned}
 & \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} \cdot \left\{ \underline{\omega}(\underline{Q}_1) \times \left[\int_V \rho(\underline{x}_1) \underline{cm}_1 dV \right] \right\} \\
 &= \int_V \rho(\underline{x}_1) \underline{A} \cdot \left\{ \underline{B} \times \underline{C} \right\} dV .
 \end{aligned}
 \tag{A-9}$$

Noting that cyclic permutations of the vectors in the scalar triple product do not change the value of the result,

$$= \int_V \rho(\underline{x}_1) \underline{B} \cdot \left\{ \underline{C} \times \underline{A} \right\} dV . \quad (\text{A-10})$$

Expressing the vector cross product in skew-symmetric form,

$$= \int_V \rho(\underline{x}_1) \underline{B}^T [\tilde{\underline{C}}] \underline{A} dV . \quad (\text{A-11})$$

Substituting the original vector functions, and moving them outside of the integral,

$$\begin{aligned} &= \int_V \rho(\underline{x}_1) \left\{ \underline{\omega}(\underline{Q}_1) \right\}^T \left[\tilde{\underline{cm}}_1 \right] \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} dV . \\ &= \int_V \rho(\underline{x}_1) dV \left\{ \underline{\omega}(\underline{Q}_1) \right\}^T \left[\tilde{\underline{cm}}_1 \right] \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} , \\ &= \left\{ \underline{\omega}(\underline{Q}_1) \right\}^T m_1 \left[\tilde{\underline{cm}}_1 \right] \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} . \end{aligned} \quad (\text{A-12})$$

Defining $\left[\underline{I}_{mL} \right] = m_1 \left[\tilde{\underline{cm}}_1 \right]^T$, this kinetic energy term can be

expressed as:

$$= \left\{ \underline{\omega}(\underline{Q}_1) \right\}^T \left[\underline{I}_{mL} \right]^T \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} \quad (\text{A-13})$$

This is the additional kinetic energy of a point mass resulting from the choice of the reference point.

The third kinetic energy term is a coupling term caused by deflections. It is similar in form to the previous term. The derivation is the same until the original vectors are substituted.

$$\begin{aligned}
 & \int_V \rho(\underline{x}_1) \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} \cdot \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{d}(\underline{x}_1) \right\} dV \\
 &= \int_V \rho(\underline{x}_1) B^T [\tilde{C}] A dV \\
 &= \int_V \rho(\underline{x}_1) \left\{ \underline{\omega}(\underline{Q}_1) \right\}^T \left[\tilde{\underline{d}}(\underline{x}_1) \right] \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} dV . \quad (A-14)
 \end{aligned}$$

Remembering that the deflections are the sum of the normal modes,

$$= \int_V \rho(\underline{x}_1) \left\{ \underline{\omega}(\underline{Q}_1) \right\}^T \sum_{j=1}^m \left[q_{1j} \left[\tilde{\underline{\delta}}_j(\underline{x}_1) \right] \right] \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} dV \quad (A-15)$$

The velocities and generalized coordinates of vibration can be moved outside of the integral:

$$= \left\{ \underline{\omega}(\underline{Q}_1) \right\}^T \sum_{j=1}^m \left[q_{1j} \int_V \rho(\underline{x}_1) \left[\tilde{\underline{\delta}}_j(\underline{x}_1) \right] dV \right] \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} . \quad (A-16)$$

Defining the integral over the skew-symmetric form of the mode shape to be

$$[\tilde{M}_{d1j}] = \int_V \rho(\underline{x}_1) \left[\tilde{\underline{\delta}}_j(\underline{x}_1) \right] dV . \quad (A-17)$$

this kinetic energy coupling term can be rewritten as

$$= \left\{ \underline{\omega}(\underline{Q}_1) \right\}^T \sum_{j=1}^m \left[[\tilde{M}_{d1j}] q_{1j} \right] \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} . \quad (A-18)$$

The fourth kinetic energy term is a coupling term between the translational velocity and the velocities of vibration.

$$\begin{aligned} & \int_V \rho(\underline{x}_1) \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} \cdot \left\{ \dot{\underline{d}}(\underline{x}_1) \right\} dV \\ &= \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\}^T \int_V \rho(\underline{x}_1) \left\{ \dot{\underline{d}}(\underline{x}_1) \right\} dV \end{aligned} \quad (A-19)$$

Substituting the mode shapes and generalized velocities of vibrations into the expression,

$$= \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\}^T \int_V \rho(\underline{x}_1) \left\{ \sum_{j=1}^m \left[\dot{q}_{1j} \underline{\delta}_j(\underline{x}_1) \right] \right\} dV \quad (A-20)$$

and taking the summation and the velocities of vibrations out of the integral, we obtain

$$= \frac{1}{2} \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\}^T \left\{ \sum_{j=1}^m \dot{q}_{1j} \left[\int_V \rho(\underline{x}_1) \underline{\delta}_j(\underline{x}_1) dV \right] \right\} . \quad (A-21)$$

Defining the integral over the mode shape:

$$\underline{M}_{d1j} = \int_V \rho(\underline{x}_1) \left[\underline{\delta}_j(\underline{x}_1) \right] dV , \quad (A-22)$$

and arranging these terms into a matrix,

$$\left[\underline{I}_{mq1} \right] = \left[\underline{M}_{d11} \mid \underline{M}_{d12} \mid \cdots \mid \underline{M}_{d1m_1} \right] , \quad (A-23)$$

the fourth kinetic energy term can be rewritten as

$$= \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\}^T \left[\underline{I}_{mq1} \right] \dot{\underline{q}}_1 . \quad (A-24)$$

The fifth kinetic energy term is the rotational kinetic energy of a rigid link.

$$\begin{aligned}
 & \frac{1}{2} \int_V \rho(\underline{x}_1) \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{x}_1 \right\} \cdot \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{x}_1 \right\} dV \\
 &= \frac{1}{2} \int_V \rho(\underline{x}_1) \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{cm}_1 \right\} \cdot \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{cm}_1 \right\} dV \\
 &+ \frac{1}{2} \int_V \rho(\underline{x}_1) \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{z}(\underline{x}_1) \right\} \cdot \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{z}(\underline{x}_1) \right\} dV \\
 &+ \left\{ \underline{\omega}(\underline{Q}_1) \times \int_V \rho(\underline{x}_1) \underline{z}(\underline{x}_1) dV \right\} \cdot \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{cm}_1 \right\} \quad (A-25)
 \end{aligned}$$

The last integral is zero because,

$$\begin{aligned}
 & \int_V \rho(\underline{x}_1) \underline{z}(\underline{x}_1) dV \\
 &= \int_V \rho(\underline{x}_1) \left\{ \underline{x}_1 - \underline{cm}_1 \right\} dV \\
 &= \int_V \rho(\underline{x}_1) \underline{x}_1 dV - \int_V \rho(\underline{x}_1) \underline{cm}_1 dV \\
 &= m_1 \underline{cm}_1 - m_1 \underline{cm}_1 = 0. \quad (A-26)
 \end{aligned}$$

The other two integrals in this kinetic energy term may be expressed in a more convenient notation:

$$\frac{1}{2} \int_V \rho(\underline{x}_1) \left\{ \underline{A} \times \underline{B} \right\} \cdot \left\{ \underline{A} \times \underline{B} \right\} dV. \quad (A-27)$$

Noting that the dot product is commutative, and the expression can be manipulated as a scalar triple product,

$$\begin{aligned}
 &= \frac{1}{2} \int_V \rho(\underline{x}_1) \left\{ \underline{A} \times \underline{B} \right\} \times \underline{A} \cdot \underline{B} \, dV \\
 &= \frac{1}{2} \int_V \rho(\underline{x}_1) \underline{A} \cdot \underline{B} \times \left\{ \underline{A} \times \underline{B} \right\} \, dV \\
 &= \frac{1}{2} \int_V \rho(\underline{x}_1) \underline{A} \cdot \left[(\underline{B} \cdot \underline{B}) \underline{A} - (\underline{B} \cdot \underline{A}) \underline{B} \right] \, dV \\
 &= \frac{1}{2} \int_V \rho(\underline{x}_1) \underline{A} \cdot \left[(\underline{B} \cdot \underline{B}) \underline{A} - \underline{B} (\underline{B} \cdot \underline{A}) \right] \, dV \\
 &= \frac{1}{2} \int_V \rho(\underline{x}_1) \underline{A}^T \left[\underline{B}^T \underline{B} \underline{A} - \underline{B} \underline{B}^T \underline{A} \right] \, dV \\
 &= \frac{1}{2} \int_V \rho(\underline{x}_1) \underline{A}^T \left[\underline{B}^T \underline{B} [\underline{I}] - \underline{B} \underline{B}^T \right] \underline{A} \, dV \tag{A-28}
 \end{aligned}$$

where $[\underline{I}]$ is the identity matrix.

Substituting the original vectors, the kinetic energy term can be expressed as

$$\begin{aligned}
 &\frac{1}{2} \int_V \rho(\underline{x}_1) \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{x}_1 \right\} \cdot \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{x}_1 \right\} \, dV \\
 &= \frac{1}{2} \int_V \rho(\underline{x}_1) \underline{\omega}(\underline{Q}_1)^T \left[\underline{cm}_1^T \underline{cm}_1 [\underline{I}] - \underline{cm}_1 \underline{cm}_1^T \right] \underline{\omega}(\underline{Q}_1) \, dV \\
 &\quad + \frac{1}{2} \int_V \rho(\underline{x}_1) \underline{\omega}(\underline{Q}_1)^T \left[\underline{\ell}(\underline{x}_1)^T \underline{\ell}(\underline{x}_1) [\underline{I}] - \underline{\ell}(\underline{x}_1) \underline{\ell}(\underline{x}_1)^T \right] \underline{\omega}(\underline{Q}_1) \, dV \tag{A-29}
 \end{aligned}$$

Notice that the term in brackets involving $\underline{\ell}$ is the three dimensional vector form of the parallel axis theorem.

Removing terms which are constant with respect to the integration, the integral in the first term yields the mass of the link.

$$\begin{aligned}
 &= \frac{1}{2} m_1 \underline{\omega}(\underline{Q}_1)^T \left[\underline{cm}_1^T \underline{cm}_1 [I] - \underline{cm}_1 \underline{cm}_1^T \right]^{(1)} \underline{\omega}(\underline{Q}_1) \\
 &+ \frac{1}{2} \underline{\omega}(\underline{Q}_1)^T \int_V \rho(\underline{x}_1) \left[\underline{x}_1^T \underline{x}_1 [I] - \underline{x}_1 \underline{x}_1^T \right] dV \underline{\omega}(\underline{Q}_1).
 \end{aligned}
 \tag{A-30}$$

The first term is the kinetic energy due to the fact that the reference point was not the center of mass and the parallel axis theorem must be used to find the equivalent inertia at the reference point. The second expression is the rotational kinetic energy of an undeflected body about the center of mass. The integral in the expression is the definition of the rigid-body rotational inertia matrix for a link. The expression can be rewritten in terms of the the rigid-body rotational inertia matrix

$$= \frac{1}{2} \underline{\omega}(\underline{Q}_1)^T \left[I_{cm/cm_1} \right] \underline{\omega}(\underline{Q}_1) + \frac{1}{2} \underline{\omega}(\underline{Q}_1)^T \left[I_{LL_1} \right] \underline{\omega}(\underline{Q}_1).
 \tag{A-31}$$

Defining the inertia about the local origin as

$$\left[I'_{LL_1} \right] = \left[I_{cm/cm_1} \right] + \left[I_{LL_1} \right],
 \tag{A-32}$$

this kinetic energy term can finally be written as

$$= \frac{1}{2} \underline{\omega}(\underline{Q}_1)^T \left[I'_{LL_1} \right] \underline{\omega}(\underline{Q}_1)
 \tag{A-33}$$

The sixth kinetic energy term is similar in form to the fifth term.

$$\int_V \rho(\underline{x}_1) \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{x}_1 \right\} \cdot \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{d}(\underline{x}_1) \right\} dV \quad (A-34)$$

Proceeding with the derivation in a similar way,

$$\begin{aligned} &= \int_V \rho(\underline{x}_1) \left\{ \underline{A} \times \underline{B} \right\} \cdot \left\{ \underline{A} \times \underline{C} \right\} dV \\ &= \int_V \rho(\underline{x}_1) \underline{A} \cdot \left[\left(\underline{C} \cdot \underline{B} \right) \underline{A} - \left(\underline{C} \cdot \underline{A} \right) \underline{B} \right] dV \\ &= \int_V \rho(\underline{x}_1) \underline{A}^T \left[\underline{C}^T \underline{B} \underline{A} - \underline{B} \underline{C}^T \underline{A} \right] dV \\ &= \int_V \rho(\underline{x}_1) \underline{A}^T \left[\underline{C}^T \underline{B} [\underline{I}] - \underline{B} \underline{C}^T \right] \underline{A} dV \end{aligned} \quad (A-35)$$

where $[\underline{I}]$ is the identity matrix. Introducing the original vector functions,

$$= \int_V \rho(\underline{x}_1) \underline{\omega}(\underline{Q}_1)^T \left[\underline{d}(\underline{x}_1)^T \underline{\underline{d}}(\underline{x}_1) [\underline{I}] - \underline{\underline{d}}(\underline{x}_1) \underline{d}(\underline{x}_1)^T \right] \underline{\omega}(\underline{Q}_1) dV \quad (A-36)$$

$\underline{\omega}(\underline{Q}_1)$ can be removed from the integral, resulting in

$$= \underline{\omega}(\underline{Q}_1)^T \int_V \rho(\underline{x}_1) \left[\underline{d}(\underline{x}_1)^T \underline{\underline{d}}(\underline{x}_1) [\underline{I}] - \underline{\underline{d}}(\underline{x}_1) \underline{d}(\underline{x}_1)^T \right] dV \quad (A-37)$$

The resulting expression is the rotational kinetic energy due to the deflection of the body. Introducing the mode shapes,

$$= \underline{\omega}(\underline{Q}_1)^T \sum_{j=1}^{m_1} q_{1j} \int \rho(\underline{x}_1) \left[\underline{\delta}_j(\underline{x}_1)^T \underline{x}_1 [I] - \underline{x}_1 \underline{\delta}_j(\underline{x}_1) \right] dV \underline{\omega}(\underline{Q}_1), \quad (A-38)$$

the integral can be defined to be

$$\left[I_{Ld1j} \right] = \int_V \rho(\underline{x}_1) \left[\underline{\delta}_j(\underline{x}_1)^T \underline{x}_1 [I] - \underline{x}_1 \underline{\delta}_j(\underline{x}_1)^T \right] dV, \quad (A-39)$$

and the sixth kinetic energy term can be rewritten as

$$= \underline{\omega}(\underline{Q}_1)^T \sum_{j=1}^{m_1} \left[\left[I_{Ld1j} \right] q_{1j} \right] \underline{\omega}(\underline{Q}_1). \quad (A-40)$$

The seventh kinetic energy term is the coupling between the rigid body rotations, and the velocities of vibration.

$$\int_V \rho(\underline{x}_1) \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{x}_1 \right\} \cdot \left\{ \dot{\underline{d}}(\underline{x}_1) \right\} dV . \quad (A-41)$$

Noting that cyclic permutations of the vectors in the scalar triple product does not change the value of the result,

$$= \int_V \rho(\underline{x}_1) \underline{\omega}(\underline{Q}_1) \cdot \left\{ \underline{x}_1 \times \dot{\underline{d}}(\underline{x}_1) \right\} dV . \quad (A-42)$$

Expressing the vector cross product in skew-symmetric form,

$$\begin{aligned} &= \int_V \rho(\underline{x}_1) \left\{ \underline{\omega}(\underline{Q}_1) \right\}^T \left[\tilde{\underline{x}}_1 \right] \left\{ \dot{\underline{d}}(\underline{x}_1) \right\} dV \\ &= \int_V \rho(\underline{x}_1) \left\{ \underline{\omega}(\underline{Q}_1) \right\}^T \left[\tilde{\underline{x}}_1 \right] \left\{ \sum_{j=1}^{m_1} \left[\dot{q}_{1j} \underline{\delta}_j(\underline{x}_1) \right] \right\} dV \\ &= \left\{ \underline{\omega}(\underline{Q}_1) \right\}^T \left\{ \sum_{j=1}^{m_1} \left[\dot{q}_{1j} \int_V \rho(\underline{x}_1) \left[\tilde{\underline{x}}_1 \right] \underline{\delta}_j(\underline{x}_1) dV \right] \right\} \end{aligned} \quad (A-43)$$

Defining the integral across the cross product as

$$\underline{M}_{Ld1j}^P = \int_V \rho(\underline{x}_1) \left[\tilde{\underline{x}}_1 \right] \underline{\delta}_j(\underline{x}_1) dV , \quad (A-44)$$

and defining a matrix,

$$[I_{Lq1}^P] = \left[\underline{M}_{Ld11}^P \mid \underline{M}_{Ld12}^P \mid \cdots \mid \underline{M}_{Ld1m_1}^P \right] , \quad (A-45)$$

this coupling term can be rewritten as

$$= \left\{ \underline{\omega}(\underline{Q}_1) \right\}^T [I_{Lq1}^P] \left\{ \dot{\underline{q}}_1 \right\} . \quad (A-46)$$

The eighth kinetic energy term is the extra rotational kinetic energy due to the deflection of the link.

$$\frac{1}{2} \int_V \rho(\underline{x}_1) \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{d}(\underline{x}_1) \right\} \cdot \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{d}(\underline{x}_1) \right\} dV \quad (A-47)$$

which is similar in form to the rigid-body kinetic energy term. The derivation is the same up to the point where the original term are introduced into the equation.

$$= \frac{1}{2} \int_V \rho(\underline{x}_1) A^T \left[B^T B [I] - B B^T \right] A dV \quad (A-48)$$

Substituting the original vectors, the kinetic energy term can be expressed as

$$\frac{1}{2} \int_V \rho(\underline{x}_1) \underline{\omega}(\underline{Q}_1)^T \left[\underline{d}(\underline{x}_1)^T \underline{d}(\underline{x}_1) [I] - \underline{d}(\underline{x}_1) \underline{d}(\underline{x}_1)^T \right] \underline{\omega}(\underline{Q}_1) dV. \quad (A-49)$$

Introducing the mode shapes results in

$$= \frac{1}{2} \underline{\omega}(\underline{Q}_1)^T \sum_{j=1}^m \sum_{k=1}^m q_{1j} q_{1k} \int_V \rho(\underline{x}_1) \left[\left[\underline{\delta}_j(\underline{x}_1) \right]^T \left[\underline{\delta}_k(\underline{x}_1) \right] - \left[\underline{\delta}_j(\underline{x}_1) \right] \left[\underline{\delta}_k(\underline{x}_1) \right]^T \right] dV \underline{\omega}(\underline{Q}_1). \quad (A-50)$$

Defining the inertia term associated with the integral as

$$\begin{aligned} \left[I_{ddijk} \right] = \int_V \rho(\underline{x}_1) & \left[\left[\underline{\delta}_j(\underline{x}_1) \right]^T \left[\underline{\delta}_k(\underline{x}_1) \right] \right. \\ & \left. - \left[\underline{\delta}_j(\underline{x}_1) \right] \left[\underline{\delta}_k(\underline{x}_1) \right]^T \right] dV . \end{aligned}$$

(A-51)

the kinetic energy term becomes

$$\frac{1}{2} \left\{ \underline{\omega}(\underline{Q}_1) \right\}^T \sum_{j=1}^{m_1} \sum_{k=1}^{m_1} \left[q_{1j} q_{1k} \left[I_{ddijk} \right] \right] \left\{ \underline{\omega}(\underline{Q}_1) \right\} .$$

(A-52)

The ninth kinetic energy term is

$$\begin{aligned}
 & \int_V \rho(\underline{x}_1) \left\{ \underline{\omega}(\underline{Q}_1) \times \underline{d}(\underline{x}_1) \right\} \cdot \left\{ \underline{\dot{d}}(\underline{x}_1) \right\} dV \\
 &= \int_V \rho(\underline{x}_1) \left\{ \underline{\omega}(\underline{Q}_1) \right\} \cdot \left\{ \underline{d}(\underline{x}_1) \times \underline{\dot{d}}(\underline{x}_1) \right\} dV \\
 &= \left\{ \underline{\omega}(\underline{Q}_1) \right\} \cdot \left\{ \int_V \rho(\underline{x}_1) \underline{\tilde{d}}(\underline{x}_1) \underline{\dot{d}}(\underline{x}_1) dV \right\} \\
 &= \left\{ \underline{\omega}(\underline{Q}_1) \right\} \cdot \left\{ \int_V \rho(\underline{x}_1) \sum_{j=1}^m \left[q_{1j} \left[\underline{\tilde{\delta}}_j(\underline{x}_1) \right] \right] \right. \\
 &\quad \left. \sum_{j=1}^m \left[\dot{q}_{1j} \underline{\delta}_j(\underline{x}_1) \right] dV \right\} \\
 &= \left\{ \underline{\omega}(\underline{Q}_1) \right\} \cdot \left\{ \sum_{j=1}^m \sum_{k=1}^m q_{1j} \int_V \rho(\underline{x}_1) \left[\underline{\tilde{\delta}}_j(\underline{x}_1) \right] \right. \\
 &\quad \left. \underline{\delta}_k(\underline{x}_1) dV \dot{q}_{1k} \right\} \quad (A-53)
 \end{aligned}$$

Defining the integral to be

$$\underline{M}_{dd1jk} = \int_V \rho(\underline{x}_1) \left[\underline{\tilde{\delta}}_j(\underline{x}_1) \right] \underline{\delta}_k(\underline{x}_1) dV, \quad (A-54)$$

this kinetic energy term can be rewritten as

$$\begin{aligned}
 &= \left\{ \underline{\omega}(\underline{Q}_1) \right\}^T \left[\sum_{j=1}^m q_{1j} \underline{M}_{dd1j1} \mid \sum_{j=1}^m q_{1j} \underline{M}_{dd1j2} \mid \dots \right. \\
 &\quad \left. \mid \sum_{j=1}^m q_{1j} \underline{M}_{dd1jm_1} \right] \dot{q}_{1k}. \quad (A-55)
 \end{aligned}$$

The tenth and final kinetic energy term is

$$\frac{1}{2} \int_V \rho(\underline{x}_1) \left\{ {}^{(1)}\dot{\underline{d}}(\underline{x}_1) \right\} \cdot \left\{ {}^{(1)}\dot{\underline{d}}(\underline{x}_1) \right\} dV . \quad (A-56)$$

Introducing the mode shapes,

$$= \frac{1}{2} \int_V \rho(\underline{x}_1) \left\{ \sum_{j=1}^{m_1} \left[\dot{q}_{1j} \underline{\delta}_j(\underline{x}_1) \right] \right\} \cdot \left\{ \sum_{k=1}^{m_1} \left[\dot{q}_{1k} \underline{\delta}_k(\underline{x}_1) \right] \right\} dV , \quad (A-57)$$

then taking the velocity of vibrations outside the integral, results in

$$= \frac{1}{2} \sum_{j=1}^{m_1} \sum_{k=1}^{m_1} \dot{q}_{1j} \dot{q}_{1k} \int_V \rho(\underline{x}_1) \left[\underline{\delta}_j(\underline{x}_1) \right] \cdot \left[\underline{\delta}_k(\underline{x}_1) \right] dV . \quad (A-58)$$

Defining the generalized mass of vibrations,

$$I_{qq1jk}^P = \int_V \rho(\underline{x}_1) \left[\underline{\delta}_j(\underline{x}_1) \right] \cdot \left[\underline{\delta}_k(\underline{x}_1) \right] dV . \quad (A-59)$$

which forms the generalized translational mass matrix of vibrations, $\left[I_{qq1}^P \right]$, the kinetic energy term can be written as

$$= \frac{1}{2} \dot{\underline{q}}_1^T \left[I_{qq1}^P \right] \dot{\underline{q}}_1 . \quad (A-60)$$

At this point it should be noted that this derivation was based on the translational mode shapes of the link. The kinetic energy contribution of the rotational modes can be obtained via a method analogous to that used for translational modes. Assuming that the local link rotational deflections are decoupled from the local link translational deflections, the rotational kinetic energy of the link can be expressed as:

$$KE_1^R = \frac{1}{2} \int_V I/V(\underline{x}_1) \left\{ \underline{\omega}(\underline{x}_1) \right\} \cdot \left\{ \underline{\omega}(\underline{x}_1) \right\} dV. \quad (A-61)$$

The angular velocity of a reference point can be written as the sum of the velocity of a reference point and the velocity relative to the reference point. The reference point is chosen to coincide with the local origin of the link. The equation of velocity is then

$$\underline{\omega}(\underline{x}_1) = \underline{\omega}(\underline{Q}_1) + \underline{\dot{\theta}}(\underline{x}_1), \quad (A-62)$$

and the rotational kinetic energy is

$$\begin{aligned} KE_1^R &= \frac{1}{2} \int_V I/V(\underline{x}_1) \left\{ \underline{\omega}(\underline{Q}_1) \right\} \cdot \left\{ \underline{\omega}(\underline{Q}_1) \right\} dV \\ &\quad + \int_V I/V(\underline{x}_1) \left\{ \underline{\omega}(\underline{Q}_1) \right\} \cdot \left\{ \underline{\dot{\theta}}(\underline{x}_1) \right\} dV \\ &\quad + \frac{1}{2} \int_V I/V(\underline{x}_1) \left\{ \underline{\dot{\theta}}(\underline{x}_1) \right\} \cdot \left\{ \underline{\dot{\theta}}(\underline{x}_1) \right\} dV. \end{aligned} \quad (A-63)$$

The first integration occurred in the previous derivation, because it is a measure of the curl of the translational velocity field. The second integration can be expressed as:

$$\begin{aligned}
 &= \int_V I/V(\underline{x}_1) \left\{ \underline{\omega}(\underline{Q}_1) \right\}^T \left\{ \sum_{j=1}^{m_1} \left[\dot{q}_{1j} \underline{\psi}_j(\underline{x}_1) \right] \right\} dV \\
 &= \left\{ \underline{\omega}(\underline{Q}_1) \right\}^T \left\{ \sum_{j=1}^{m_1} \left[\dot{q}_{1j} \int_V I/V(\underline{x}_1) \underline{\psi}_j(\underline{x}_1) dV \right] \right\} \quad (A-64)
 \end{aligned}$$

Defining the integral to be

$$\underline{M}_{Ld1j}^R = \int_V I/V(\underline{x}_1) \underline{\psi}_j(\underline{x}_1) dV \quad (A-65)$$

and defining a matrix

$$[I_{Lq1}^R] = \left[\underline{M}_{Ld11}^R \mid \underline{M}_{Ld12}^R \mid \cdots \mid \underline{M}_{Ld1m_1}^R \right], \quad (A-66)$$

this coupling term can be rewritten as

$$= \left\{ \underline{\omega}(\underline{Q}_1) \right\} [I_{Lq1}^R] \left\{ \dot{\underline{g}}_1 \right\}. \quad (A-67)$$

The third, and final integration can be expressed as

$$= \frac{1}{2} \int_V I/V(\underline{x}_1) \left\{ \sum_{j=1}^{m_1} \left[\dot{q}_{1j} \underline{\psi}_j(\underline{x}_1) \right] \right\} \cdot \left\{ \sum_{k=1}^{m_1} \left[\dot{q}_{1k} \underline{\psi}_k(\underline{x}_1) \right] \right\} dV \quad (A-68)$$

Taking the velocity of vibrations outside the integral,

$$= \frac{1}{2} \sum_{j=1}^{m_1} \sum_{k=1}^{m_1} \dot{q}_{1j} \dot{q}_{1k} \int_V I/V(\underline{x}_1) \left[\underline{\psi}_j(\underline{x}_1) \right] \cdot \left[\underline{\psi}_k(\underline{x}_1) \right] dV . \quad (A-69)$$

Next, defining the generalized mass of vibrations,

$$I_{qq1jk}^R = \int I/V(\underline{x}_1) \left[\underline{\psi}_j(\underline{x}_1) \right] \cdot \left[\underline{\psi}_k(\underline{x}_1) \right] dV \quad (A-70)$$

which forms the rotational generalized mass matrix of vibrations, $\left[I_{qq1}^R \right]$. The final integration can be expressed as

$$= \frac{1}{2} \dot{\underline{g}}_1^T \left[I_{qq1}^R \right] \dot{\underline{g}}_1 . \quad (A-71)$$

The kinetic energy can now be presented in terms of the new mass and inertia matrices.

Defining:

$$\begin{bmatrix} I_{Lq_1} \end{bmatrix} = \begin{bmatrix} I_{Lq_1}^P \end{bmatrix} + \begin{bmatrix} I_{Lq_1}^R \end{bmatrix} \quad (A-72)$$

and

$$\begin{bmatrix} I_{qq_1} \end{bmatrix} = \begin{bmatrix} I_{qq_1}^P \end{bmatrix} + \begin{bmatrix} I_{qq_1}^R \end{bmatrix}, \quad (A-73)$$

the total kinetic energy is

$$\begin{aligned} KE_1 = & \frac{1}{2} \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\}^T \begin{bmatrix} I_{mm_1} \end{bmatrix} \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} \\ & + \left\{ \underline{\omega}(\underline{Q}_1) \right\}^T \begin{bmatrix} I_{mL_1} \end{bmatrix}^T \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\} \\ & + \left\{ \dot{\underline{R}}(\underline{Q}_1) \right\}^T \begin{bmatrix} I_{mq_1} \end{bmatrix} \left\{ \dot{\underline{q}}_1 \right\} \\ & + \frac{1}{2} \left\{ \underline{\omega}(\underline{Q}_1) \right\}^T \begin{bmatrix} I_{LL_1} \end{bmatrix} \left\{ \underline{\omega}(\underline{Q}_1) \right\} \\ & + \left\{ \underline{\omega}(\underline{Q}_1) \right\}^T \begin{bmatrix} I_{Lq_1} \end{bmatrix} \left\{ \dot{\underline{q}}_1 \right\} \\ & + \frac{1}{2} \left\{ \dot{\underline{q}}_1 \right\}^T \begin{bmatrix} I_{qq_1} \end{bmatrix} \left\{ \dot{\underline{q}}_1 \right\} \\ & + \delta(KE) + \delta^2(KE) \end{aligned} \quad (A-74)$$

where $\delta(KE)$ and $\delta^2(KE)$ are the kinetic energy term which are functions of the link deflections, and are small.

This expression can be organized into the desired quadratic form.

$$KE_1 = \frac{1}{2} \dot{\underline{\phi}}_1^T \left[\underline{I}_1 \right] \dot{\underline{\phi}}_1 \quad (A-75)$$

$$= \frac{1}{2} \begin{Bmatrix} \dot{\underline{R}}(\underline{Q}_1) \\ \dot{\underline{\omega}}(\underline{Q}_1) \\ \dot{\underline{q}}_1 \end{Bmatrix}^T \begin{bmatrix} \left[\underline{I}_{mm_1} \right] & \left[\underline{I}_{mL_1} \right] & \left[\underline{I}_{mq_1} \right] \\ \left[\underline{I}_{mL_1} \right]^T & \left[\underline{I}_{LL_1} \right] & \left[\underline{I}_{Lq_1} \right] \\ \left[\underline{I}_{mq_1} \right]^T & \left[\underline{I}_{Lq_1} \right]^T & \left[\underline{I}_{qq_1} \right] \end{bmatrix} \begin{Bmatrix} \dot{\underline{R}}(\underline{Q}_1) \\ \dot{\underline{\omega}}(\underline{Q}_1) \\ \dot{\underline{q}}_1 \end{Bmatrix}$$

(A-76)

This local inertia matrix is used in the derivation of the equations of motion. In deriving these inertia terms, the reader must be cautious that none of the assumptions or restrictions of the vibrations model are violated. Due to the generality of this modeling method, these restrictions are not inherently included as the equations were derived. Those unfamiliar with the restrictions which apply to models of vibrations in continuous systems are urged to refer to Thomson [23].

APPENDIX B

Derivation of the Stiffness Matrix

The expression of the potential energy due to elastic deflection of a system is given as:

$$PE_1 = \frac{1}{2} \int_V EI/V(\underline{x}_1) \partial^2/\partial \underline{x}_1^2 \left\{ {}^{(1)}\underline{d}(\underline{x}_1) \right\} \cdot \partial^2/\partial \underline{x}_1^2 \left\{ {}^{(1)}\underline{d}(\underline{x}_1) \right\} dV . \quad (B-1)$$

In this case, the link is assumed to be long and thin. The equation for the local deflection is assumed to be the sum of the normal modes:

$$\underline{d}(\underline{x}_1) = \sum_{j=1}^m q_{1j} {}^{(1)}\underline{\delta}_{1j} \quad (B-2)$$

such that

$$PE_1 = \frac{1}{2} \int_V EI/V(\underline{x}_1) \sum_{j=1}^m q_{1j} \partial^2/\partial \underline{x}_1^2 \left\{ {}^{(1)}\underline{\delta}_{1j} \right\} \cdot \sum_{j=1}^m q_{1j} \partial^2/\partial \underline{x}_1^2 \left\{ {}^{(1)}\underline{\delta}_{1j} \right\} dV . \quad (B-3)$$

The summation and the generalized coordinates of vibrations can be taken outside the integral,

$$PE_1 = \frac{1}{2} \sum_{j=1}^m \sum_{k=1}^m q_{1j} q_{1k} \int_V EI/V(\underline{x}_1) \partial^2/\partial \underline{x}_1^2 \left\{ {}^{(1)}\underline{\delta}_{1j} \right\} \cdot \partial^2/\partial \underline{x}_1^2 \left\{ {}^{(1)}\underline{\delta}_{1k} \right\} dV . \quad (B-4)$$

Now, define the integration to be the stiffness term relating the potential energy due to the deflections q_{1j} and q_{1k} .

$$K_{1jk}^P = \int_V EI/V(\underline{x}_1) \partial^2/\partial \underline{x}_1^2 \left\{ {}^{(1)}\underline{\delta}_{1j} \right\} \cdot \partial^2/\partial \underline{x}_1^2 \left\{ {}^{(1)}\underline{\delta}_{1k} \right\} dV . \quad (B-5)$$

analogously for the rotational deflections:

$$K_{1jk}^R = \int_V GJ/V(\underline{x}_1) \partial/\partial \underline{x}_1 \left\{ {}^{(1)}\underline{\psi}_{1j} \right\} \cdot \partial/\partial \underline{x}_1 \left\{ {}^{(1)}\underline{\psi}_{1k} \right\} dV . \quad (B-6)$$

The potential energy can now be expressed in quadratic form,

$$PE_i = \frac{1}{2} \underline{g}_i^T \left[\underline{K}_i \right] \underline{g}_i . \quad (B-7)$$

$$\text{where } \left[\underline{K}_i \right] = \left[\underline{K}_i^P \right] + \left[\underline{K}_i^R \right] \quad (B-8)$$

APPENDIX C

VSIm: A Simulation Package

The source code for the simulation resides on the Silicon Graphics 4-D computer system in the University of Texas Mechanical Systems Robotics Lab. It may be found in the following directory:

```
/usr/people/philip/VIBES
```

To perform a simulation, the following files must be altered for the system which is to be simulated.

```
/usr/people/philip/VIBES/Robotdat.c
```

and

```
/usr/people/philip/VIBES/Model.c
```

The first contains the mass, geometry, and stiffness data for the system to be simulated. The control model for the system must be included in the second file. It is suggested that the user be familiar with the screen editor Vi, which is the common editor in a Unix system. The user must also be able to program in the computer language "C" to implement a control model. The whole program must be compiled by the following command:

```
cc -O3 -Zg -o VSIm VSIm.c
```

This compiles the program using the full optimization option.

To execute the simulation, type:

VSim outfile

The word "outfile" refers to the name of the file in which the output data will be found. It may be replaced by any other name, and may include a directory, using the common Unix format. The program will ask what the starting and ending times should be, and what the minimum, and maximum time steps should be.

```

#include "math.h"
#include "stdio.h"

#include "Robotdat.c"
#include "External.c"

#include "Const.c"
#include "Cmath.c"
#include "Gear.c"
#include "Geometry.c"
#include "Kinematics.c"
#include "Istar.c"
#include "Kstar.c"
#include "Torque.c"
#include "Model.c"

main(argc,argv)

int argc;
char *argv[];

double endtime,hh,hhmin,hhmax,eps,printtime,oldprint;
double t,y[8][order],save[12][order],ymax[order];
double error[order],pw[order][order],told,hhold,hhnew,*aa;
int i,j,mf,kflag,jstart,maxder,nq,nqold,newq,k,numdata;
FILE *dat_file,*fopen();

if(argc>1) {

    printf("The output filename is called %s\n\n",*(argv+1));

    printf("starttime= ?\n");
    scanf("%lf",&t);
    printf("starttime= %e\n",t);
    printf("endtime= ?\n");
    scanf("%lf",&endtime);
    printf("endtime= %e\n",endtime);
    printf("how many data points ?\n");
    scanf("%d",&numdata);
    printf("initial step size= ?\n");
    scanf("%lf",&hh);
    printf("initial step size= %e\n",hh);
    printf("minimum step size= ?\n");
    scanf("%lf",&hhmin);
    printf("minimum step size= %e\n",hhmin);
    printf("maximum step size= ?\n");
    scanf("%lf",&hhmax);
    printf("maximum step size= %e\n",hhmax);
    printf("error constant= ?\n");
    scanf("%lf",&eps);
    printf("error constant= %e\n",eps);

    dat_file=fopen(*(argv+1),"w");
    for(i=0;i<order;i++) y[0][i]=0.;

    mf=0;
    for(i=0;i<order;i++) {

```



```

        ymax[i]=1.0;
        error[i]=0.0;
    }
    jstart=0;
    maxder=6;
    printtime=(endtime-t)/numdata;
    oldprint=t;

    printf("The integration is now starting \n");
    Gear(&t,y,save,&hh,&hhmin,&hhmax,eps,mf,ymax,error,&kflag,&jstart,
        maxder,pw,&told,&shhold,&shhnew,&nq,&nqold,&newq,&k);

    jstart=1;
    fprintf(dat_file,"%d\n",dof);
    while(t<endtime && kflag>0) {
        printf("%d",nq);
        if(t>(oldprint+printtime)) {
            oldprint=t;
            fprintf(dat_file,"%e ",t);
            for(j=0;j<dof;j++) fprintf(dat_file,"%1.4e ",y[0][j]);
            fprintf(dat_file,"\n");
            Gear(&t,y,save,&hh,&hhmin,&hhmax,eps,mf,ymax,error,&kflag,
                &jstart,maxder,pw,&told,&shhold,&shhnew,&nq,&nqold,&newq,&k);
        }
        fclose(dat_file);
        printf("kflag=%d\n",kflag);
    }

    else printf("Please include the name of the output file \n");

```

ORIGINAL PAGE IS
OF POOR QUALITY

```
#define order  dof*2  
typedef double Vector[3], Matrix[3][3];  
double sa[n], ca[n], stheta[n], ctheta[n];
```

```

Const(G,Gp)

/* INITIALIZE THE JACOBIAN AND HESSIAN MATRICIES TO ALL ZERO */
/* COMPUTE CONSTANT VALUES IN THE JACOBIAN AND HESSIAN MATRICIES */

double G[n][6][dof], Gp[n][6][dof];

int i,j,k,l;
extern int state_type[n], state_dir[dof], state_pos[n], mt[n], mr[n];
extern int jflag[n];
extern double alpha[n], theta[n], sa[n], ca[n], stheta[n], ctheta[n];

/* COMPUTE THE TRANSCENDENTAL FUNCTIONS OF THE CONSTANT ANGLE
LINK PARAMETERS */
for(i=0;i<n;i++) {
    sa[i]=sin(alpha[i]);
    ca[i]=cos(alpha[i]);
    stheta[i]=sin(theta[i]);
    ctheta[i]=cos(theta[i]);
}

/* INITIALIZE THE JACOBIANS */
for(j=0;j<n;j++) {
    for(k=0;k<6;k++) {
        for(l=0;l<dof;l++) G[j][k][l]=0.;
        for(l=0;l<dof;l++) Gp[j][k][l]=0.;
    }
}

/* ASSEMBLE ALL CONSTANT FLEX DIRECTION COSINES */
i=h;
for(j=h-1;j<n;j++) {
    l=state_pos[h]+jflag[h]+1;
    for(k=1;k<1+mt[i];k++)
        Gp[j][state_dir[k]][k]=1.; /* translational flex */
    for(;k<state_pos[h+1];k++)
        Gp[j][state_dir[k]+3][k]=1.; /* rotational flex */
}

/* ASSEMBLE ALL CONSTANT JOINT DIRECTION COSINES */
for(j=h-1;j>=0;j--) {
    G[j][5-3*state_type[h]][state_pos[h]]=1.0;
    Gp[j][5-3*state_type[h]][state_pos[h]]=1.0;
}

/* ASSEMBLE ALL CONSTANT JOINT FLEX DIRECTION COSINES */
if(jflag[h]) {
    for(j=h-1;j>=0;j--) {
        G[j][5-3*state_type[h]][state_pos[h]+1]=1.0;
        Gp[j][5-3*state_type[h]][state_pos[h]+1]=1.0;
    }
}

/* ALL GROSS MOTION DIRECTION COSINES */
if (/free) {
    for(j=0;j<n;j++) {
        for(l=6;l>0;l--) G[j][6-l][dof-l]=1.;
        for(l=6;l>0;l--) Gp[j][6-l][dof-l]=1.;
    }
}

```

```

#define          max_ele      dof*2

/*****
/* PRINT MATRIX TO SCREEN */
*****/

PRINTM(M,i,j)

double *M;
int i,j;

int k,l,row;

printf("\n");
for (k=0;k<i;k++)
{
    row=k*j;
    printf("\n");
    if(*(M+row+1)>=0)
        for(l=0;l<j;l++) printf(" %1.2e ",*(M+row+1));
    else
        for(l=0;l<j;l++) printf("%1.2e ",*(M+row+1));
    printf("\n");
}
printf("\n");
return(1);

*****/
* MATRIX MULTIPLY */
*****/

MM(M,i,j,N,k,l,O)

double *M,*N,*O;
int i,j,k,l;

int e,f,g;

if (j!=k) return(-1); /* check to see if matrices are conformable */
for (e=0;e<i;e++) /* row counter for matrix M */
{
    for (f=0;f<l;f++) /* column counter for matrix N */
    {
        *(O+l*e-f)=0.0;
        for (g=0;g<j;g++) *(O+l*e-f)+= *(M+j*e-g) * *(N-l*g+f);
    }
}

return(1);

*****/
* TRANSPOSE */
*****/

T(M,i,j,N)

int i,j;

```

```

double *M, *N;

int k,l;
for (k=0;k<i;k++)
    {
        for(l=0;l<j;l++) *(N+l*i+k) = *(M+k*j+l);
    }
return(l);
}

/*****
 * SCALAR MULTIPLY */
*****/

SM(M,i,j,S,N)

int i,j;
double *M, *N, S;

int k,l;
for (k=0;k<i;k++)
    {
        for(l=0;l<j;l++)
            {
                *(N+k*j+l) = S * *(M+k*j+l);
            }
    }

return(l);

/*****
 * CROSS PRODUCTS */
*****/

CrossX(t,r,G)

Matrix t,G;
Vector r;

g[0][0]=t[1][0]*r[2]-t[2][0]*r[1];
g[0][1]=t[2][0]*r[0]-t[0][0]*r[2];
g[0][2]=t[0][0]*r[1]-t[1][0]*r[0];
return(l);

CrossY(t,r,G)

Matrix t,G;
Vector r;

g[0][1]=t[1][1]*r[2]-t[2][1]*r[1];
g[1][1]=t[2][1]*r[0]-t[0][1]*r[2];
g[2][1]=t[0][1]*r[1]-t[1][1]*r[0];
return(l);

CrossZ(t,r,G)

```

```

Matrix t,G;
Vector r;

{
G[0][2]=t[1][2]*r[2]-t[2][2]*r[1];
G[1][2]=t[2][2]*r[0]-t[0][2]*r[2];
G[2][2]=t[0][2]*r[1]-t[1][2]*r[0];
return(1);
}

/*
 * GENERATE A ROTATION MATRIX */
/*
*****

ROT(axis,radians,M)

int axis;
double radians;
Matrix M;

{
int i,j;
for(i=0;i<3;i++)
    for(j=0;j<3;j++)
        M[i][j]=0.0;

if(axis==2)
    M[0][0]=cos(radians);
    M[0][1]=-sin(radians);
    M[1][0]=sin(radians);
    M[1][1]=cos(radians);
    M[2][2]=1.0;

else if(axis==1)
    M[0][0]=cos(radians);
    M[0][2]=sin(radians);
    M[1][1]=1.0;
    M[2][0]=-sin(radians);
    M[2][2]=cos(radians);

else if(axis==0)
    M[0][0]=1.0;
    M[1][1]=cos(radians);
    M[1][2]=-sin(radians);
    M[2][1]=sin(radians);
    M[2][2]=cos(radians);

return(1);
}

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

/*****
/*  MATRIX INVERSE  */
/*    using        */
/*  Gaussian-Jordan */
/*    Elimination   */
*****/

GInverse(ti,a,nn)

/* ti points to the matrix to be inverted */
/* a points to the inverted matrix */
/* nn is the dimension of the matrix */

double *a, *ti;
int nn;

{
int ipiv[max_ele], indxr[max_ele], indxc[max_ele];
int i, j, k, irow, icol, l, ll, row, col;
double big,dum,pivinv;

for (i=0;i<nn;i++) {
    row=i*nn;
    for (j=0;j<nn;j++) {
        *(a+row+j) = *(ti+row+j);
    }
}

for(j=0;j<nn;j++) *(ipiv+j)=0;

for(i=0;i<nn;i++) {
    big=0.;
    for(j=0;j<nn;j++) {
        row=j*nn;
        if(*(ipiv+j)!=1) {
            for(k=0;k<nn;k++) {
                if(*(ipiv+k) == 0) {
                    if(fabs(*(a+row+k))>=big) {
                        big=fabs(*(a+row+k));
                        irow=j;
                        icol=k;
                    }
                }
            }
            else if(*(ipiv+k)>1) {
                printf("singular matrix");
                return;
            }
        }
    }

    -- *(ipiv+icol);
    row=nn*irow;
    col=nn*icol;
    if(irow!=icol) {
        for(l=0;l<nn;l++) {
            dum = *(a+row+l);
            *(a+row+l) = *(a-col+l);
            *(a-col+l)=dum;
        }
    }
}

```

```

    }
    *(indx+1)=irow;
    *(indxc+1)=icol;
    if(*(a+col+icol)==0) {
        printf("singular matrix");
        return;
    }
    pivinv=1./ *(a+col+icol);
    *(a+col+icol)=1.;
    for(l=0;l<nn;l++) *(a+col+l)= *(a+col+l)*pivinv;
    for(ll=0;ll<nn;ll++) {
        if(ll!=icol) {
            row=nn+ll;
            col=nn+icol;
            dum= *(a+row+icol);
            *(a+row+icol)=0.;
            for(l=0;l<nn;l++) *(a+row+l) -= *(a+col+l)*dum;
        }
    }
}
for(l=nn-1;l>=0;l--) {
    if(*(indx+1) != *(indxc+1)) {
        for(k=0;k<nn;k++) {
            row=nn+k;
            dum= *(a+row+ *(indx+1));
            *(a+row+ *(indxc+1))= *(a+row+ *(indx+1));
            *(a+row+ *(indxc+1))=dum;
        }
    }
}
return;

```



```
Geometry(phi,t,tl,r)
```

```

* GIVEN THE GEOMETRY STATES, CREATE THE TRANSFORMATION MATRICIES AND
  THE POSITION VECTORS */
/* NOTE THAT THE CONSTANT POSITION VECTOR r[h], THE CONSTANT TRANS-
  FORMATION MATRIX t[h], THE TRANSCENDENTALS OF ALPHA[n] AND THETA[n]
  ARE ALREADY ASSIGNED VALUES BY THE CONSTANTS ROUTINE */

```

```

double phi[dof];
Matrix tl[n],t[n];
Vector r[n];

```

```

*
  phi[dof] : THE GEOMETRY STATES
  t[n][3][3] : THE TRANSFORMATION FROM LOCAL SYSTEM i TO SYSTEM H
  tl[n][3][3] : THE LOCAL TRANSFORMATION MATRIX FROM i TO i-1 WHEN
                  i>h, OR FROM i TO i+1 WHEN i<h.
  r[n][3] : THE POSITION VECTOR FROM ORIGIN H TO ORIGIN N

```

```

/*
extern int state_type[n], state_pos[n], state_dir[dof], mt[n], mr[n], h;
extern double mode[dof], sa[n], ca[n], theta[n], stheta[n], ctheta[n];
extern Vector L[n];
*/

```

```

int i,j,k,l,m;
double a,s,c;
Vector del[n], rdel[n], r1,r2;
Matrix tl;

```

```

* THE ROTATION MATRIX ASSOCIATED WITH FRAME h IS THE IDENTITY MATRIX */
for(i=0;i<3;i++) {
  for(j=0;j<3;j++) {
    tl[h][i][j]=0.;
    t[h][i][j]=0.;
  }
  tl[h][i][i]=1.;
  t[h][i][i]=1.;
}

```

```

* COMPUTE LOCAL DEFLECTIONS OF EACH LINK */
for(i=0;i<n;i++) {
  j=state_pos[i]-jflag[i]+1;

```

```

  * INITIALIZE THE DEFLECTIONS TO ZERO */
  for(k=0;k<3;k++) {
    del[i][k]=0;
    rdel[i][k]=0;
  }

```

```

  /* SUM THE TRANSLATIONAL DEFLECTIONS OF LINK i */
  /* k DENOTES THE STATE POSITION ASSOCIATED WITH A DEFLECTION
    OF LINK i */
  for (k=j;k<j+mt[i];k++) del[i][state_dir[k]]+=phi[k]*mode[k];

```

```

  /* SUM THE ROTATIONAL DEFLECTIONS OF LINK i */
  /* k DENOTES THE STATE POSITION ASSOCIATED WITH A DEFLECTION

```

```

      OF LINK i */
      for (;k<j+mt[i]+mr[i];k++) rdel[i][state_dir[k]]+=phi[k]*mode[k];
    }

/* COMPUTE 3x3 ROTATION MATRICIES AND POSITION VECTORS FOR ALL > h */
for(i=h+1; i<n; i++) {
  j=state_pos[i];

  /* COMPUTE TRANSCENDENTAL FUNCTIONS OF JOINT i */
  /* IF JOINT i IS ROTATIONAL, INCLUDE JOINT i MAGNITUDE
  AND DEFLECTION (theta[i] is the constant joint rotation) */
  if(!state_type[i]) {
    c=cos(phi[j]-phi[j+1]*jflag[i]+theta[i]);
    s=sin(phi[j]-phi[j+1]*jflag[i]+theta[i]);
  }

  /* IF JOINT i IS TRANSLATIONAL, OR CONSTRAINED, LEAVE OUT
  JOINT i MAGNITUDE AND DEFLECTION, AND INCLUDE CONSTANT
  LINK i PARAMETER TRANSCENDENTALS */
  else {
    c=ctheta[i];
    s=stheta[i];
  }

  /* FORM LOCAL TRANSFORMATION MATRIX TO TRANSFORM A VECTOR FROM
  LOCAL FRAME i TO LOCAL FRAME i-1. THIS TRANSFORMATION IS A
  FUNCTION OF THE TRANSCENDENTALS OF JOINT i AND ALPHA i-1,
  AND LINK i-1 DEFLECTIONS */
  k=i-1;
  t1[i][0][0]=c-rdel[k][2]*ca[k]*s-rdel[k][1]*sa[k]*s;
  t1[i][0][1]=-s-rdel[k][2]*ca[k]*c-rdel[k][1]*sa[k]*c;
  t1[i][0][2]=rdel[k][2]*sa[k]+rdel[k][1]*ca[k];
  t1[i][1][0]=rdel[k][2]*c-ca[k]*s-rdel[k][0]*sa[k]*s;
  t1[i][1][1]=-rdel[k][2]*s-ca[k]*c-rdel[k][0]*sa[k]*c;
  t1[i][1][2]=-sa[k]-rdel[k][0]*ca[k];
  t1[i][2][0]=-rdel[k][1]*c-rdel[k][0]*ca[k]*s+sa[k]*s;
  t1[i][2][1]=rdel[k][1]*s-rdel[k][0]*ca[k]*c-sa[k]*c;
  t1[i][2][2]=-rdel[k][0]*sa[k]+ca[k];

  /* FORM THE TRANSFORMATION MATRIX TO TRANSFORM A VECTOR FROM
  LOCAL FRAME i TO GLOBAL FRAME h */
  t[t[i-1][3,3],t1[i][3,3],t[i]];

  /* FORM THE LOCAL POSITION VECTOR FROM ORIGIN i-1 TO ORIGIN i
  THIS VECTOR IS A FUNCTION OF THE LENGTH OF LINK i-1, THE
  DEFLECTION OF LINK i-1, THE TRANSLATION OF JOINT i, AND THE
  TRANSLATIONAL DEFLECTION OF JOINT i. THIS LOCAL POSITION
  VECTOR IS IN THE i-1 LOCAL FRAME */
  /* IF JOINT i IS TRANSLATIONAL, INCLUDE JOINT i MAGNITUDE AND
  DEFLECTION */
  if(state_type[i]) {
    m=state_pos[i];
    for(k=0; k<3; k++)
      rl[k]=L[i-1][k]+del[i-1][k]
      +t[i][k][2]*(phi[m]-phi[m+1]*jflag[i]);
  }

  /* IF JOINT i IS ROTATIONAL, LEAVE OUT JOINT i MAGNITUDE AND
  DEFLECTION */
  else
    for(k=0; k<3; k++) rl[k]=L[i-1][k]+del[i-1][k];
}

```

```

/* CONVERT LOCAL POSITION VECTOR TO FRAME H */
MM(t[i-1],3,3,r1,3,1,r2);
/* FORM THE POSITION VECTOR FROM ORIGIN H TO ORIGIN I */
for(k=0;k<3;k++) r[i][k]=r[i-1][k]+r2[k];
}

/* COMPUTE 3x3 ROTATION MATRICIES AND POSITION VECTORS FOR ALL < H */
for(i=h;i>0;i--) {
    j=state_pos[i];

    /* COMPUTE TRANSCENDENTAL FUNCTIONS OF JOINT i */
    /* IF JOINT i IS ROTATIONAL, INCLUDE JOINT i MAGNITUDE
    AND DEFLECTION (theta[i] is the constant joint rotation) */
    if(!state_type[i]) {
        c=cos(phi[j]+phi[j+1]*jflag[i]+theta[i]);
        s=sin(phi[j]+phi[j+1]*jflag[i]+theta[i]);
    }
    /* IF JOINT i IS TRANSLATIONAL, LEAVE OUT JOINT i MAGNITUDE
    AND DEFLECTION, AND INCLUDE CONSTANT LINK i PARAMETER
    TRANSCENDENTALS */
    else {
        c=ctheta[i];
        s=stheta[i];
    }

    /* FORM LOCAL TRANSFORMATION MATRIX TO TRANSFORM A VECTOR FROM
    LOCAL FRAME i-1 TO LOCAL FRAME i. THIS TRANSFORMATION IS A
    FUNCTION OF THE TRANSCENDENTALS OF JOINT i AND ALPHA i-1,
    AND LINK i-1 DEFLECTIONS */
    k=i-1;
    tl[k][0][0]=c-rdel[k][2]*ca[k]*s+rdel[k][1]*sa[k]*s;
    tl[k][1][0]=-s-rdel[k][2]*ca[k]*c-rdel[k][1]*sa[k]*c;
    tl[k][2][0]=rdel[k][2]*sa[k]-rdel[k][1]*ca[k];
    tl[k][0][1]=rdel[k][2]*c+ca[k]*s-rdel[k][0]*sa[k]*s;
    tl[k][1][1]=-rdel[k][2]*s+ca[k]*c-rdel[k][0]*sa[k]*c;
    tl[k][2][1]=-sa[k]-rdel[k][0]*ca[k];
    tl[k][1][2]=-rdel[k][1]*c+rdel[k][0]*ca[k]*s+sa[k]*s;
    tl[k][2][2]=rdel[k][1]*s+rdel[k][0]*ca[k]*c+sa[k]*c;
    tl[k][3][2]=-rdel[k][0]*sa[k]+ca[k];

    /* FORM THE TRANSFORMATION MATRIX TO TRANSFORM A VECTOR FROM
    LOCAL FRAME i-1 TO GLOBAL FRAME h */
    MM(t[i],3,3,tl[k],3,3,t[k]);

    /* FORM THE LOCAL POSITION VECTOR FROM ORIGIN i TO ORIGIN i-1
    THIS VECTOR IS A FUNCTION OF THE LENGTH OF LINK i-1, THE
    DEFLECTION OF LINK i-1, THE TRANSLATION OF JOINT i, AND THE
    TRANSLATIONAL DEFLECTION OF JOINT i */
    /* IF JOINT i IS TRANSLATIONAL, INCLUDE JOINT i MAGNITUDE AND
    DEFLECTION */
    if(!state_type[i]) {
        m=state_pos[i];
        for(k=0;k<3;k++)
            r1[k]=-L[i-1][k]-del[i-1][k]
                -t[i][k][2]*(phi[m]+phi[m+1]*jflag[i]);
    }
    /* IF JOINT i IS ROTATIONAL, LEAVE OUT JOINT i MAGNITUDE AND
    DEFLECTION */
    else {

```

```

        for(k=0;k<3;k++) r1[k]= -L[i-1][k]-del[i-1][k];
    }

    /* CONVERT LOCAL POSITION VECTOR TO FRAME H */
    MM(t[i-1],3,3,r1,3,1,r2);

    /* FORM THE POSITION VECTOR FROM ORIGIN H TO ORIGIN i-1 */
    for(k=0;k<3;k++) r[i-1][k]=r[i][k]+r2[k];
}

```

```

Kinematics(t,r,G,Gp)

/* GIVEN THE TRANSFORMATION MATRICIES t[n] AND THE POSITION VECTORS
r[n], THE JACOBIAN AND HESSIAN MATRICIES FOR EACH LOCAL FRAME ARE FORMED */

Matrix t[n];
Vector r[n];
double G[n][6][dof],Gp[n][6][dof];

int i, j, k, l, ll, ll2, ll3;
Matrix GPJKP[n][n], GPJKD[n][n], GPJK[n][n];
Vector Fl;
extern int state_type[n], state_dir[dof], state_pos[n], mt[n], mr[n];

* COMPUTE ALL CROSS PRODUCTS WHICH ARE THE GEOMETRIC INFLUENCE
OF THE ROTATIONS ON THE TRANSLATIONS OF THE LOCAL FRAMES */

* COMPUTE THE CROSS PRODUCTS ASSOCIATED WITH THE GROSS MOTION
OF LINK h */
for(i=0;i<h;i++) {
    Crossx(t[h],r[i],GPJK[h][i]);
    Crossy(t[h],r[i],GPJK[h][i]);
    Crossz(t[h],r[i],GPJK[h][i]);
}
for(i=h+1;i<n;i++) {
    Crossx(t[h],r[i],GPJK[h][i]);
    Crossy(t[h],r[i],GPJK[h][i]);
    Crossz(t[h],r[i],GPJK[h][i]);
}

* COMPUTE CROSS PRODUCTS ASSOCIATED WITH JOINT MOTIONS AND VIBRATIONS */

* FOR ALL LINKS, DEFINE THE PROXIMAL END OF LINK i TO
COINCIDE WITH THE LOCAL REFERENCE FRAME i, AND THE DISTAL END
OF THE LINK TO COINCIDE WITH LOCAL REFERENCE FRAME i+1 */

* NOTE THAT FOR LINKS NUMBERED > h, THE MOTION OF JOINT i CONTRIBUTES
TO THE MOTION OF LOCAL FRAMES i, i+1, ... n-1. BUT THE VIBRATIONS OF LINK i
(WHICH IS DEFINED AS THE MOTION OF FRAME i+1 RELATIVE TO THE MOTION
OF JOINT i) CONTRIBUTES TO THE MOTION OF LOCAL FRAMES i+1, i+2, ... n-1.
THEREFORE THE CROSS PRODUCTS ASSOCIATED WITH THE MOTION OF JOINT i
WILL BE DIFFERENT THAN THE CROSS PRODUCTS ASSOCIATED WITH THE
VIBRATIONS OF LINK i. THE CROSS PRODUCTS ASSOCIATED WITH JOINT
MOTION WILL BE CALLED GPJKP, AND THE CROSS PRODUCTS ASSOCIATED WITH
VIBRATIONS WILL BE CALLED GPJKD. */

* i DENOTES THE JOINT VELOCITY OR LINK VIBRATIONS REFERENCED TO THE
ith LOCAL FRAME */
for(i=h+1;i<(n-1);i++) {
    /* j DENOTES THE LOCAL FRAME WHERE THE POSITION VECTOR TERMINATES */
    for(j=i+1;j<=n;j++) {
        /* COMPUTE CROSS PRODUCT OF LOCAL LINK i JOINT AXIS AND
        POSITION VECTORS ORIGINATING AT THE PROXIMAL END OF LINK

```

```

i+1 AND TERMINATING AT THE ORIGIN OF LINK j. THIS IS
THE CROSS PRODUCT ASSOCIATED WITH JOINT MOTION i */
for(k=0;k<3;k++) r1[k]=r[j][k]-r[i][k];
Crossz(t[i],r1,GPJKP[i][j]);
}

for(j=i+2;j<=n;j++) {

/* COMPUTE CROSS PRODUCT OF LOCAL LINK i AXES AND
POSITION VECTORS ORIGINATING AT THE DISTAL END OF
LINK i+1 AND TERMINATING AT THE ORIGIN OF LINK j.
THIS IS THE CROSS PRODUCT ASSOCIATED WITH THE
THE VIBRATIONS OF LINK i */

for(k=0;k<3;k++) r1[k]=r[j][k]-r[i][k];

/* THE CROSS PRODUCT IN THE JOINT DIRECTION IS */
for(k=0;k<3;k++)
    GPJKD[i][j][k]=GPJKP[i][j][k][2]-GPJKP[i][i+1][k][2];

/* THE CROSS PRODUCTS IN THE X AND Y DIRECTIONS ARE */
Crossx(t[i],r1,GPJKD[i][j]);
Crossy(t[i],r1,GPJKD[i][j]);
}

}

* THE h JOINT AXIS IS A SPECIAL CASE, WHICH EFFECTS THE LINKS NUMBERED < h */
* THE CROSS PRODUCTS ASSOCIATED WITH THIS HAVE BEEN COMPUTED, ONLY THE SIGN
MUST BE CHANGED WHEN THIS CROSS PRODUCT IS USED FOR JOINT h MOTION */

* NOTE THAT FOR LINKS NUMBERED < h, THE MOTION OF JOINT i CONTRIBUTES
TO THE MOTION OF LOCAL FRAMES i-1, ... 0, AND THE VIBRATIONS OF LINK i
WHICH IS DEFINED AS THE MOTION OF FRAME i+1 RELATIVE TO THE MOTION
OF JOINT i) CONTRIBUTES TO THE MOTION OF LOCAL FRAMES i, i-1, ... 0.
BUT, THE CROSS PRODUCTS ASSOCIATED WITH THE MOTION OF JOINT i
WILL BE THE SAME AS THE CROSS PRODUCTS ASSOCIATED WITH THE
VIBRATIONS OF LINK i, SINCE THEIR RELATIVE MOTIONS COINCIDE AT THE
LOCAL ith COORDINATE FRAME. */

* i DENOTES THE JOINT VELOCITY OR LINK VIBRATIONS REFERENCED TO THE
LOCAL FRAME */
for(i=n-1;i>0;i--) {

/* j DENOTES THE LOCAL FRAME WHERE THE POSITION VECTOR TERMINATES */
for(j=i-1;j>=0;j--) {

/* COMPUTE CROSS PRODUCT OF LOCAL LINK i AXES AND
POSITION VECTORS ORIGINATING AT THE DISTAL END OF
LINK i-1 AND TERMINATING AT THE ORIGIN OF LINK j.
THIS IS THE CROSS PRODUCT ASSOCIATED WITH BOTH
THE JOINT MOTION i AND THE VIBRATIONS OF LINK i */

for(k=0;k<3;k++) r1[k]=r[i][k]-r[j][k];
Crossx(t[i],r1,GPJK[i][j][0]);
Crossy(t[i],r1,GPJK[i][j][1]);
Crossz(t[i],r1,GPJK[i][j][2]);
}
}

```

```

/* ASSEMBLE THE JACOBIANS */

/* ALL DATA NEEDED TO FORM THE JACOBIAN IS NOW AVAILABLE FROM THE
TRANSFORMATION MATRICIES, AND THE CROSS PRODUCTS THAT WERE COMPUTED */

/* AT THE BEGINNING OF THE PROGRAM, ALL n JACOBIANS WERE INITIALIZED TO
ZERO, AND THE CONSTANT PARTS, ASSOCIATED WITH THE GROSS MOTION, THE
TRANSLATIONAL AND ROTATIONAL DEFLECTIONS OF LINK h (EXCEPT THE CROSS
PRODUCTS ASSOCIATED WITH THE ROTATIONS), AND THE MOTION OF JOINT h
(EXCEPT THE CROSS PRODUCTS ASSOCIATED WITH A ROTATIONAL JOINT), ARE
COMPUTED. THE JACOBIAN FOR FRAME h IS ALWAYS CONSTANT. ONLY THE TIME-
VARYING PARTS OF THE JACOBIANS ARE CHANGED IN THIS ROUTINE */

* j DENOTES THE JACOBIAN FOR FRAME j */

* i INDICATES THAT THE JOINT MOTION OR VIBRATION IS ASSOCIATED WITH LINK i */

* k DENOTES THE STATE VECTOR POSITION OF JOINT MOTION, LINK VIBRATION,
OR GROSS MOTION. THE CONVENTION USED IS k EQUALS:

    if i=0 (link 0):
        state_pos[0]...state_pos[mt[0]-1]
            = translational deflections of link 0.
        state_pos[mt[0]]...state_pos[mt[0]+mr[0]-1]
            = rotational deflections of link 0.

    if i=1...n-1 (all other joints and links):
        state_pos[i] = motion of joint i.
        if jflag = 1 (if there is flex in joint i)
            state_pos[i]+jflag[i] = deflection of joint i.
        if mt[i]>0 (if there is translational flex in link i)
            state_pos[i]+jflag[i]+1...state_pos[i]+jflag[i]-mt[i]
                = translational deflections of link i.
        if mr[i]>0 (if there is rotational flex in link i)
            state_pos[i]+jflag[i]+1+mt[i]...state_pos[i]+jflag[i]-mt[i]-mr[i]
                = rotational deflections of link i.

    state_pos[dof]-6 = gross translation of link h in the x direction
    state_pos[dof]-5 = gross translation of link h in the y direction
    state_pos[dof]-4 = gross translation of link h in the z direction
    state_pos[dof]-3 = gross rotation of link h in the x direction
    state_pos[dof]-2 = gross rotation of link h in the y direction
    state_pos[dof]-1 = gross rotation of link h in the z direction

* FOR j > h */

/* THE NON-CONSTANT JOINT DIRECTION COSINES */
* i DENOTES JOINT NUMBER */
for(i=h-1;i<n;i++) {
    ll=state_pos[i];
    /* IF THIS JOINT HAS FLEX INCLUDE IT IN THE JACOBIAN */
    if(jflag[i]) {
        /* l2 IS THE STATE POSITION OF THE OF JOINT i FLEX */
        l2=ll-1;
        /* j DENOTES JACOBIAN OF LOCAL FRAME j */
    }
}

```

```

for(j=i;j<n;j++) {
    /* JOINT POSITION PARTS */
    /* 1 IS ZERO IF TRANSLATIONAL, OR THREE
       IF ROTATIONAL JOINT */
    l = 3-3*state_type[i];
    G[j][1][11] = t[i][0][2];
    G[j][1+1][11] = t[i][1][2];
    G[j][1+2][11] = t[i][2][2];
    Gp[j][1][11] = t[i][0][2];
    Gp[j][1+1][11] = t[i][1][2];
    Gp[j][1+2][11] = t[i][2][2];
    /* JOINT FLEX PARTS */
    G[j][1][12] = t[i][0][2];
    G[j][1+1][12] = t[i][1][2];
    G[j][1+2][12] = t[i][2][2];
    Gp[j][1][12] = t[i][0][2];
    Gp[j][1+1][12] = t[i][1][2];
    Gp[j][1+2][12] = t[i][2][2];
}

/* IF THIS JOINT HAS NO FLEX, LEAVE IT OUT OF THE JACOBIAN */
else {
    /* j DENOTES JACOBIAN OF LOCAL FRAME j */
    for(j=i;j<n;j++) {
        /* JOINT POSITION PARTS */
        /* 1 IS ZERO IF TRANSLATIONAL, OR THREE
           IF ROTATIONAL JOINT */
        l = 3-3*state_type[i];
        G[j][1][11] = t[i][0][2];
        G[j][1+1][11] = t[i][1][2];
        G[j][1+2][11] = t[i][2][2];
        Gp[j][1][11] = t[i][0][2];
        Gp[j][1+1][11] = t[i][1][2];
        Gp[j][1+2][11] = t[i][2][2];
    }
}

/* ALL NON-CONSTANT JOINT CROSS-PRODUCTS */
/* 1 DENOTES JOINT NUMBER */
for(i=h-1;i<n-1;i++) {
    /* CHECK TO SEE IF THE JOINT IS ROTATIONAL */
    if(!state_type[i]) {
        l1 = state_pos[i];
        /* IF THIS JOINT HAS FLEX, INCLUDE IT IN THE JACOBIAN */
        if (jflag[i]) {
            l2=l1+1;
            /* j DENOTES JACOBIAN OF LOCAL FRAME j */
            for(j=i-1;j<n;j++) {
                /* JOINT POSITION PARTS */
                G[j][0][11] = GPJKP[i][j][0][2];
                G[j][1][11] = GPJKP[i][j][1][2];
                G[j][2][11] = GPJKP[i][j][2][2];
                Gp[j][0][11] = GPJKP[i][j][0][2];
                Gp[j][1][11] = GPJKP[i][j][1][2];
                Gp[j][2][11] = GPJKP[i][j][2][2];
                /* JOINT FLEX PARTS */
                G[j][0][12] = GPJKP[i][j][0][2];
                G[j][1][12] = GPJKP[i][j][1][2];
                G[j][2][12] = GPJKP[i][j][2][2];
                Gp[j][0][12] = GPJKP[i][j][0][2];
            }
        }
    }
}

```



```

Gp[j][1][12] = GPJKP[i][j][1][2];
Gp[j][2][12] = GPJKP[i][j][2][2];
}

/* IF THIS JOINT HAS NO FLEX, LEAVE IT OUT OF THE JACOBIAN */
else {
/* j DENOTES JACOBIAN OF LOCAL FRAME j */
for(j=i+1;j<n;j++) {
/* JOINT POSITION PARTS */
G[j][0][11] = GPJKP[i][j][0][2];
G[j][1][11] = GPJKP[i][j][1][2];
G[j][2][11] = GPJKP[i][j][2][2];
Gp[j][0][11] = GPJKP[i][j][0][2];
Gp[j][1][11] = GPJKP[i][j][1][2];
Gp[j][2][11] = GPJKP[i][j][2][2];
}
}

/* GROSS MOTION CROSS-PRODUCTS */
/* j DENOTES JACOBIAN OF LOCAL FRAME j */
if(free) {
for(j=h+1;j<n;j++) {
/* x-DIRECTION */
G[j][0][dof-3] = GPJK[h][j][0][0];
G[j][1][dof-3] = GPJK[h][j][1][0];
G[j][2][dof-3] = GPJK[h][j][2][0];
Gp[j][0][dof-3] = GPJK[h][j][0][0];
Gp[j][1][dof-3] = GPJK[h][j][1][0];
Gp[j][2][dof-3] = GPJK[h][j][2][0];
/* y-DIRECTION */
G[j][0][dof-2] = GPJK[h][j][0][1];
G[j][1][dof-2] = GPJK[h][j][1][1];
G[j][2][dof-2] = GPJK[h][j][2][1];
Gp[j][0][dof-2] = GPJK[h][j][0][1];
Gp[j][1][dof-2] = GPJK[h][j][1][1];
Gp[j][2][dof-2] = GPJK[h][j][2][1];
/* z-DIRECTION */
G[j][0][dof-1] = GPJK[h][j][0][2];
G[j][1][dof-1] = GPJK[h][j][1][2];
G[j][2][dof-1] = GPJK[h][j][2][2];
Gp[j][0][dof-1] = GPJK[h][j][0][2];
Gp[j][1][dof-1] = GPJK[h][j][1][2];
Gp[j][2][dof-1] = GPJK[h][j][2][2];
}

/* THE NON-CONSTANT FLEX DIRECTION COSINES */
/* i DENOTES LINK ASSOCIATED WITH THIS FLEX */
for(i=h-1;i<n-1;i++) {
l1=state_pos[i]+jflag[i]+1;
l2=l1-mt[i];
l3=l2-mr[i];
/* j DENOTES JACOBIAN OF LOCAL FRAME j */
for(j=i-1;j<n;j++) {
/* k DENOTES STATE POSITION OF TRANSLATIONAL LINK FLEX
AND THE COLUMN OF THE JACOBIAN */
for(k=l1;k<l2;k++) {
Gp[j][0][k] = t[i][0][state_dir[k]];
Gp[j][1][k] = t[i][1][state_dir[k]];
}
}
}

```

```

        Gp[j][2][k] = t[i][2][state_dir[k]];
    }
    /* k DENOTES STATE POSITION OF ROTATIONAL LINK FLEX
    AND THE COLUMN OF THE JACOBIAN */
    for(;k<13;k++) {
        Gp[j][3][k] = t[i][0][state_dir[k]];
        Gp[j][4][k] = t[i][1][state_dir[k]];
        Gp[j][5][k] = t[i][2][state_dir[k]];
    }
}

* THE NON-CONSTANT FLEX CROSS PRODUCTS */
* i DENOTES LINK ASSOCIATED WITH THIS FLEX */
for(i=h;i<n-2;i++) {
    l1=state_pos[i]+jflag[i]+mt[i];
    l2=l1+mr[i];
    /* j DENOTES JACOBIAN OF LOCAL FRAME j */
    for(j=i+2;j<n;j++) {
        /* k DENOTES STATE POSITION OF ROTATIONAL LINK FLEX
        AND THE COLUMN OF THE JACOBIAN */
        for(k=11;k<12;k++) {
            Gp[j][0][k] = GPJKD[i][j][0][state_dir[k]];
            Gp[j][1][k] = GPJKD[i][j][1][state_dir[k]];
            Gp[j][2][k] = GPJKD[i][j][2][state_dir[k]];
        }
    }
}

* FOR j < h */
* THE NON-CONSTANT FLEX DIRECTION COSINES */
* i DENOTES LINK ASSOCIATED WITH THIS FLEX */
for(i=h-1;i>=0;i--) {
    l1=state_pos[i]-jflag[i]-1;
    l2=l1-mt[i];
    l3=l2-mr[i];
    /* j DENOTES JACOBIAN OF LOCAL FRAME j */
    for(j=1;j>=0;j--) {
        /* k DENOTES STATE POSITION OF TRANSLATIONAL LINK FLEX
        AND THE COLUMN OF THE JACOBIAN */
        for(k=11;k<12;k++) {
            G[j][0][k] = -t[i][0][state_dir[k]];
            G[j][1][k] = -t[i][1][state_dir[k]];
            G[j][2][k] = -t[i][2][state_dir[k]];
        }
        /* k DENOTES STATE POSITION OF ROTATIONAL LINK FLEX
        AND THE COLUMN OF THE JACOBIAN */
        for(;k<13;k++) {
            G[j][3][k] = -t[i][0][state_dir[k]];
            G[j][4][k] = -t[i][1][state_dir[k]];
            G[j][5][k] = -t[i][2][state_dir[k]];
        }
    }
}

* THE NON-CONSTANT FLEX CROSS PRODUCTS */
* i DENOTES LINK ASSOCIATED WITH THIS FLEX */
for(i=h-1;i>=0;i--) {
    l1=state_pos[i]+jflag[i]+1+mt[i];
    l2=l1-mr[i];
    /* j DENOTES JACOBIAN OF LOCAL FRAME j */

```

```

for(j=i-1;j>=0;j--) {
  /* k DENOTES STATE POSITION OF ROTATIONAL LINK FLEX
  AND THE COLUMN OF THE JACOBIAN */
  for(k=l1;k<l2;k++) {
    G[j][0][k] = GPJK[i][j][0][state_dir[k]];
    G[j][1][k] = GPJK[i][j][1][state_dir[k]];
    G[j][2][k] = GPJK[i][j][2][state_dir[k]];
  }
}

/* THE NON-CONSTANT JOINT DIRECTION COSINES */
/* i DENOTES JOINT NUMBER */
for(i=n-1;i>0;i--) {
  l1=state_pos[i];
  /* IF THERE IS JOINT FLEX, INCLUDE IT IN THE JACOBIAN */
  if (jflag[i]) {
    l2=l1+1;
    /* j DENOTES JACOBIAN OF LOCAL FRAME j */
    for(j=i-1;j>=0;j--) {
      /* JOINT POSITION PARTS */
      /* 1 IS ZERO IF TRANSLATIONAL, OR THREE
      IF ROTATIONAL JOINT */
      l = 3-j*state_type[i];
      G[j][1][l1] = -t[i][0][2];
      G[j][1+1][l1] = -t[i][1][2];
      G[j][1+2][l1] = -t[i][2][2];
      /* JOINT FLEX PARTS */
      G[j][1][l2] = -t[i][0][2];
      G[j][1+1][l2] = -t[i][1][2];
      G[j][1+2][l2] = -t[i][2][2];
    }
  }
  /* IF THERE IS NO JOINT FLEX, LEAVE IT OUT OF THE JACOBIAN */
  else {
    /* j DENOTES JACOBIAN OF LOCAL FRAME j */
    for(j=i-1;j>=0;j--) {
      /* JOINT POSITION PARTS */
      /* 1 IS ZERO IF TRANSLATIONAL, OR THREE
      IF ROTATIONAL JOINT */
      l = 3-j*state_type[i];
      G[j][1][l1] = -t[i][0][2];
      G[j][1+1][l1] = -t[i][1][2];
      G[j][1+2][l1] = -t[i][2][2];
    }
  }
}

/* ALL NON-CONSTANT JOINT CROSS-PRODUCTS */
/* i DENOTES JOINT NUMBER */
i=0;
/* CHECK TO SEE IF THE JOINT IS ROTATIONAL */
if(!state_type[i]) {
  l1=state_pos[i];
  /* IF THERE IS JOINT FLEX, INCLUDE IT IN THE JACOBIAN */
  if (jflag[i]) {
    l2=l1+1;
    /* j DENOTES JACOBIAN OF LOCAL FRAME j */
    for(j=i-1;j>=0;j--) {
      /* JOINT POSITION PARTS */
      G[j][0][l1] = -GPJK[i][j][0][2];
    }
  }
}

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

G[j][1][11] = -GPJK[i][j][1][2];
G[j][2][11] = -GPJK[i][j][2][2];
/* JOINT FLEX PARTS */
G[j][0][12] = -GPJK[i][j][0][2];
G[j][1][12] = -GPJK[i][j][1][2];
G[j][2][12] = -GPJK[i][j][2][2];
}
}
/* IF THERE IS NO JOINT FLEX, LEAVE IT OUT OF THE JACOBIAN */
else {
/* j DENOTES JACOBIAN OF LOCAL FRAME j */
for(j=i-1;j>=0;j--) {
/* JOINT POSITION PARTS */
G[j][0][11] = -GPJK[i][j][0][2];
G[j][1][11] = -GPJK[i][j][1][2];
G[j][2][11] = -GPJK[i][j][2][2];
}
}
}
for(i=h-1;i>0;i--) {
/* CHECK TO SEE IF THE JOINT IS ROTATIONAL */
if(state_type[i]) {
l1=state_pos[i];
/*IF THERE IS JOINT FLEX, INCLUDE IT IN THE JACOBIAN */
if(jflag[i]) {
l2=l1-1;
/* j DENOTES JACOBIAN OF LOCAL FRAME j */
for(j=i-1;j>=0;j--) {
/* JOINT POSITION PARTS */
G[j][0][11] = GPJK[i][j][0][2];
G[j][1][11] = GPJK[i][j][1][2];
G[j][2][11] = GPJK[i][j][2][2];
/* JOINT FLEX PARTS */
G[j][0][12] = GPJK[i][j][0][2];
G[j][1][12] = GPJK[i][j][1][2];
G[j][2][12] = GPJK[i][j][2][2];
}
}
/* IF THERE IS NO JOINT FLEX, LEAVE IT OUT OF THE JACOBIAN */
else {
/* j DENOTES JACOBIAN OF LOCAL FRAME j */
for(j=i-1;j>=0;j--) {
/* JOINT POSITION PARTS */
G[j][0][11] = GPJK[i][j][0][2];
G[j][1][11] = GPJK[i][j][1][2];
G[j][2][11] = GPJK[i][j][2][2];
}
}
}
}
}
/* GROSS MOTION CROSS-PRODUCTS */
/* j DENOTES JACOBIAN OF LOCAL FRAME j */
if (free) {
for(j=h-1;j>=0;j--) {
/* x-DIRECTION */
G[j][0][dof-3]=GPJK[h][j][0][0];
G[j][1][dof-3]=GPJK[h][j][1][0];
G[j][2][dof-3]=GPJK[h][j][2][0];
/* y-DIRECTION */
G[j][0][dof-2]=GPJK[h][j][0][1];

```

```
G(j)[1][dof-2]=GPJK(h)(j)[1][1];
G(j)[2][dof-2]=GPJK(h)(j)[2][1];
/* z-DIRECTION */
G(j)[0][dof-1]=GPJK(h)(j)[0][2];
G(j)[1][dof-1]=GPJK(h)(j)[1][2];
G(j)[2][dof-1]=GPJK(h)(j)[2][2];
)
```

```

istar(istr,istrc,g,t)

double istr[n-1][dof][dof], istrc[n+1][dof][dof];
double g[n][6][dof], t[n][3][3];

int i,j,k,row,col,cl;
double a[dof][dof], b[dof][dof], c[dof][dof], al,b1,c1,d1;

/* i DENOTES THE FRAME OF LINK (OR LOCAL REFERENCE) NUMBER */
/* COMPUTE THE istar OF EACH LINK SEPARATELY, AND ADD THEM UP
   AT THE END OF THE ROUTINE TO FORM istar FOR THE SYSTEM */

for(i=0;i<n;i++) {

    /* INITIALIZE THE istar MATRIX FOR LINK i */
    for(row=0;row<dof;row++)
        for(col=0;col<dof;col++)
            istr[i][row][col]=0.;

    /* FIRST, FORM THE UPPER-RIGHT SYMMETRIC PARTS OF
       THE SUB-MATRICES ON THE DIAGONAL OF THE LOCAL INERTIA
       MATRIX, THEN FORM THE LOWER-LEFT PARTS.
       NEXT FORM THE OFF DIAGONAL SUB-MATRICES IN THE UPPER-
       RIGHT AREA OF THE LOCAL INERTIA MATRIX, AND TRANSPOSE
       THEM TO GET THE LOWER-LEFT PARTS. */

    /* TRANSLATIONAL MASS */
    /* PRE-MULTIPLY THE TRANSLATIONAL JACOBIAN BY ITS TRANSPOSE
       AND MULTIPLY BY THE MASS */
    for(row=0;row<dof;row++)
        for(col=0;col<dof;col++) {
            al=m[i]*(g[i][0][row]*g[i][0][col]
                    +g[i][1][row]*g[i][1][col]
                    +g[i][2][row]*g[i][2][col]);
            istr[i][row][col]+=al;
        }

    /* RIGID BODY ROTATION */
    /* UPPER-RIGHT TERMS, TAKING ADVANTAGE OF SYMMETRY */
    /* COMPUTE INERTIA IN GLOBAL FRAME */
    for(col=0;col<3;col++) {

        al= I11[i][0][0]*t[i][col][0]
            -I11[i][0][1]*t[i][col][1]
            -I11[i][0][2]*t[i][col][2];

        b1= I11[i][1][0]*t[i][col][0]
            -I11[i][1][1]*t[i][col][1]
            -I11[i][1][2]*t[i][col][2];

        c1= I11[i][2][0]*t[i][col][0]
            -I11[i][2][1]*t[i][col][1]
            -I11[i][2][2]*t[i][col][2];

        for(row=col;row<3;row++) {
            a[row][col]= t[i][row][0]*al
                        +t[i][row][1]*b1
                        +t[i][row][2]*c1;

```

```

        a[col][row] = a[row][col];
    }

/* PRE- AND POST- MULTIPLY THE INERTIA BY THE ROTATIONAL
   JACOBIAN */
for(col=0; col<dof; col++) {

    a1= a[0][0]*g[i][3][col]
        +a[0][1]*g[i][4][col]
        +a[0][2]*g[i][5][col];

    b1= a[1][0]*g[i][3][col]
        +a[1][1]*g[i][4][col]
        +a[1][2]*g[i][5][col];

    c1= a[2][0]*g[i][3][col]
        +a[2][1]*g[i][4][col]
        +a[2][2]*g[i][5][col];

    for(row=0; row<dof; row++) {
        d1=g[i][3][row]*a1
            +g[i][4][row]*b1
            +g[i][5][row]*c1;

        istr[i][row][col] += d1;
    }

/* ADD THE LINK FLEXIBILITY INERTIA TERMS TO istr */
/* (NO TRANSFORMATIONS ARE NECESSARY) */
j=nt[i]-mr[i];
k=state_pos[i]-jflag[i]-1;
for(row=0; row<j; row++)
    for(col=0; col<j; col++)
        istr[i][row+k][col-k] += Iqq[i][row][col];

/* RIGID-BODY/ROTATIONAL COUPLING */
/* AN OFF-DIAGONAL MATRIX */
/* COMPUTE INERTIA IN GLOBAL FRAME */
for(col=0; col<3; col++) {

    a1= Iml[i][0][0]*t[i][col][0]
        -Iml[i][0][1]*t[i][col][1]
        -Iml[i][0][2]*t[i][col][2];

    b1= Iml[i][1][0]*t[i][col][0]
        -Iml[i][1][1]*t[i][col][1]
        -Iml[i][1][2]*t[i][col][2];

    c1= Iml[i][2][0]*t[i][col][0]
        -Iml[i][2][1]*t[i][col][1]
        -Iml[i][2][2]*t[i][col][2];

    for(row=col; row<3; row++) {
        a[row][col] = t[i][row][0]*a1
            +t[i][row][1]*b1
            +t[i][row][2]*c1;

        a[col][row] = a[row][col];
    }
}

```

```

    }
/* POST MULTIPLY BY THE ROTATIONAL JACOBIAN */
for(row=0;row<3;row++)
    for(col=0;col<dof;col++)
        b[row][col]= a[row][0]*g[i][3][col]
                    +a[row][1]*g[i][4][col]
                    +a[row][2]*g[i][5][col];

/* PRE MULTIPLY BY THE TRANSLATIONAL JACOBIAN */
for(row=0;row<dof;row++)
    for(col=0;col<dof;col++)
        c[row][col]= g[i][0][row]*b[0][col]
                    +g[i][1][row]*b[1][col]
                    +g[i][2][row]*b[2][col];

/* ADD THIS MATRIX AND ITS TRANSPOSE TO istr */
for(row=0;row<dof;row++)
    for(col=0;col<dof;col++) {
        istr[i][row][col]+=c[row][col];
        istr[i][col][row]+=c[row][col];
    }

/* TRANSLATION/FLEX COUPLING */
/* MULTIPLY THE INERTIA BY THE TRANSFORMATION MATRIX */
j=nt[i]+nr[i];
for(row=0;row<3;row++)
    for(col=0;col<j;col++)
        a[row][col]= t[i][row][0]*Imq[i][0][col]
                    -t[i][row][1]*Imq[i][1][col]
                    -t[i][row][2]*Imq[i][2][col];

/* MULTIPLY BY THE TRANSLATIONAL JACOBIAN */
for(row=0;row<dof;row++)
    for(col=0;col<j;col++)
        b[row][col]= g[i][0][row]*a[0][col]
                    -g[i][1][row]*a[1][col]
                    -g[i][2][row]*a[2][col];

/* ADD THIS MATRIX AND ITS TRANSPOSE TO istr */
k=state_pos[i]-jflag[i]-1;
for(row=0;row<dof;row++)
    for(col=k;col<k+j;col++) {
        istr[i][row][col]-=b[row][col-k];
        istr[i][col-k][row]-=b[row][col-k];
    }

/* ROTATION/FLEX COUPLING */
/* MULTIPLY THE INERTIA MATRIX BY THE TRANSFORMATION MATRIX */
j=nt[i]+nr[i];
for(row=0;row<3;row++)
    for(col=0;col<j;col++)
        a[row][col]= t[i][row][0]*Ilq[i][0][col]
                    -t[i][row][1]*Ilq[i][1][col]
                    +t[i][row][2]*Ilq[i][2][col];

/* MULTIPLY BY THE TRANSLATIONAL JACOBIAN */
for(row=0;row<dof;row++)
    for(col=0;col<j;col++)
        b[row][col]= g[i][3][row]*a[0][col]
                    -g[i][4][row]*a[1][col]

```



```

                                +g[i][5][row]*a[2][col];

/* ADD THIS MATRIX AND ITS TRANSPOSE TO istr */
k=state_pos[i]+jflag[i]+1;
for(row=0;row<dof;row++)
    for(col=k;col<k+j;col++) {
        istr[i][row][col]+=b[row][col-k];
        istr[i][col][row]+=b[row][col-k];
    }
}

* ADD THE CONTRIBUTIONS OF EACH LINK TO FORM THE GROSS Istar */
for(row=0;row<dof;row++)
    for(col=0;col<dof;col++) {
        istr[n][row][col]=0.;
        istr[n][col][row]=0.;
        for(i=0;i<n;i++)
            istr[n][row][col]+=istr[i][row][col];
            istr[n][col][row]+=istr[i][col][row];
    }
}

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

Kstar(kstr)

double kstr[dof][dof];

int i,k,l,row,col;

for(i=0;i<n;i++) {
    k=state_pos[i]+jflag[i];
    if(jflag[i]==1) kstr[k][k]=kjoint[i];
    k++;
    l=mt[i]+mr[i];
    if(l!=0)
        for(row=0;row<l;row++)
            for(col=0;col<l;col++)
                kstr[k+row][k+col]=klink[i][row][col];
}

```

```
Torque(tque,load,G)
double tque[dof],load[n][6],G[n][6][dof];
{
int i,j;
double a;
for(i=0;i<dof;i++) {
    a=0.;
    for(j=0;j<6;j++) {
        a+=load[n-1][j]*G[n-1][j][i];
    }
    tque[i]=lmode[i]*a;
}
```

ORIGINAL PAGE IS
OF POOR QUALITY

```

Model(hh,time,y,save)

double *hh,*time,y[8][order],*save;

double phi[dof],disturb;
Matrix tl[n],t[n];
Vector r[n];
double endload[n][6],load[dof],jointload[dof],tque[dof];
double G[n][6][dof],Gp[n][6][dof];
double istr[n+1][dof][dof],istrinv[dof][dof];
double kstr[dof][dof],istrc[n+1][dof][dof];
double pstr[n+1][dof][dof];
int i,j;

for(i=0;i<6;i++)
    for(j=0;j<n;j++) {
        endload[j][i]=0.;
    }

for(i=0;i<dof;i++) phi[i]=y[0][i];
Const(G,Gp);
Geometry(phi,t,tl,r);
Kinematics(t,r,G,Gp);
Istar(istr,istrc,Gp,t);
/* Pstar(G,H,SkewG,pstr); */
Kstar(kstr);

/* APPLY END LOAD */
if( *time>.3 && *time<2.) endload[n-1][0]=1000.;
Torque/tque,endload,Gp);

/* INVERT THE INERTIA MATRIX */
Inverse(&istr[n][0][0],istrinv,dof);

/* APPLY JOINT TORQUES */
if(*time<3.) {
    jointload[0]=0.;
    jointload[1]=0.;
    jointload[2]=0.;
    jointload[3]=0.;
    jointload[4]=0.;
    jointload[5]=0.;

/* COMPUTE LOAD VECTOR AND ASSEMBLE RIGID BODY INERTIA */
for(i=0;i<dof;i++) {
    load[i]=tque[i]+jointload[i];
    for(j=0;j<dof;j++) {
        a=0.;
        for(k=0;k<dof;k++) a+=y[0][dof-k]*pstar[j][i][k];
        load[i]-=a*y[0][dof+j];

        load[i]-=kstr[i][j]*y[0][j];
        load[i]-=(alpha*istr[n][i][j]+beta*fabs(kstr[i][j]))*y[0][dof-j];
        load[i]-=istrc[n][i][j]* y[1][dof+j]/ *hh;
    }
}

```

```

for(i=0;i<dof;i++) *(save+i)=y[0][i+dof];
for(;i<order;i++) {
    *(save+i)=0.;
    for(j=0;j<dof;j++)
        *(save+i)+=(istrinv[i-dof][j]*load[j]);
}

/* APPLY BREAKS IF DESIRED */
for(i=1;i<7;i++) {
    if(brakes[i]) {
        /* SET VELOCITY = 0 */
        *(save+state_pos[i])=0.0;

        /* SET ACCELERATION = 0 */
        *(save+dof+state_pos[i])=0.0;
    }
}

```

ORIGINAL PAGE IS
OF POOR QUALITY

REFERENCES

1. Benedict, C.E., and Tesar, D., "Dynamic Response Analysis of Quasi-static Mechanical Systems Using Kinematic Influence Coefficients, " Transactions of the ASME, Journal of Mechanisms, Vol. 6, 1971, pp. 383-403.
2. Benedict, C.E., and Tesar, D., "Model Formulation of Complex Mechanisms with Multiple inputs: Part I - Geometry," Transactions of the ASME, Journal of Mechanical Design, October, 1978, pp. 747-750.
3. Benedict, C.E., and Tesar, D., "Model Formulation of Complex Mechanisms with Multiple inputs: Part II - The Dynamic Model," Transactions of the ASME, Journal of Mechanical Design, October, 1978, pp. 751-761.
4. Thomas, M., and Tesar, D., "Dynamic Modeling of Serial Manipulator Arms," Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control, Vol. 104, September, 1982, pp. 218-228.
5. Freeman, R., and Tesar, D., "Kinematic and Dynamic Modeling, Analysis and Control of Robotic Mechanisms (Via Generalized Coordinate Transformation)," Manufacturing Systems Engineering, Department of Mechanical Engineering, The University of Texas at Austin, June 1985; also Ph.D. dissertation by Freeman, University of Florida, Gainesville, August 1985.
6. Fresonke, D. A., and Tesar, D., "Deflection Prediction for Quasi-static Serial Manipulators," Manufacturing Systems Engineering, Department of Mechanical Engineering, The University of Texas at Austin 1985, also Masters Thesis by Fresonke, 1985.

7. Behi, F., and Tesar, D., "Dynamic Analysis and Parametric Identification for Flexible Serial Manipulators," Manufacturing Systems Engineering, Department of Mechanical Engineering, The University of Texas at Austin, 1985, also Ph.D dissertation by Behi, The University of Florida, Gainesville, 1985
8. Wander, J. P., and Tesar, D., "Pipelined Computation of Manipulator Modeling Matrices," IEEE Journal of Robotics and Automation, Vol. RA-3, No. 6, December, 1987, pp. 556-566.
9. Denavit, J., and Hartenberg, R. S., "A Kinematic Notation for Lower Pair Mechanisms Based on Matrices," Transactions of the ASME, Journal of Applied Mechanics, Vol. 22, No. 3, June, 1955, pp. 215-221.
10. Hollerbach, J. M., "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity," IEEE Trans. on Systems, Man, and Cybernetics, Vol. 11, November 1980, pp. 730-736.
11. Book, W. J., "Modeling, Design and Control of Flexible Manipulator Arms," Ph.D. Thesis, Massachusetts Institute of Technology Department of Mechanical Engineering. 1974
12. Book, W. J., Maizza-Neto, O., and Whitney, D. E., "Feedback Control of Two Beam, Two Joint Systems with Distributed Flexibility," Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control, 97G(4):424-431, December 1975.
13. Book, W. J., "Analysis of Massless Elastic Chains with Servo Controlled Joints," Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control, 101(3):187-192, September 1979.

14. Centinkunt, S., and Book, W. J., "Symbolic Modeling of Flexible Manipulators," Proceedings of the 1987 IEEE International Conference on Robotics and Automation, Vol.3, pp. 2074-2080.
15. Whitney, D. E., Book, W. J., and Lynch, P. M., "Design and Control Considerations for Industrial and Space Manipulators," Proceedings of the 1974 Joint Automatic Controls Conference, AIChE, pp 591-598, 1974
16. Hughes, P. C., "Dynamics of a Chain of Flexible Bodies," Journal of the Astronautical Sciences, Vol. XXVII, No. 4, 359-380, October-December, 1979
17. Low, K. H., "A Systematic Formulation of Dynamic Equations for Robotic Manipulators with Elastic Links," Journal of Robotic Systems, 4(3), 435-456, 1987.
18. Sunada, W. H., and Dubowsky, S., "On the Dynamic Analysis and Behavior of Industrial Robotic Manipulators With Elastic Members," Journal of Mechanisms, Transmissions, and Automation in Design, Vol 105, pp 42-51, 1983
19. Naganathan, G., Soni, A. H., "Coupling Effects of Kinematics and Flexibility in Manipulators," The International Journal of Robotics Research, Vol. 6, No. 1, pp 75-84, Spring, 1987a.
20. Huang, Y., Lee, C. S. G., "Generalization of Newton-Euler Formulation of Dynamic Equations to Nonrigid Manipulators," Proceedings of the 1987 American Control Conference, Vol. 1, pp 72-77, June, 1987.
21. King, J. O., Gourishankar, V. G., Rink, R. E., "Lagrangian Dynamics of Flexible Manipulators Using Angular Velocities Instead of Transformation Matrices," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-17, No. 11, November-December, 1987

22. Sanders, J. R., and Tesar, D., "The Analytical and Experimental Evaluation of Vibratory Oscillations in Realistically Proportioned Mechanisms," Transactions of the ASME, Journal of Mechanical Design, Vol. 100, October 1978, 762-768.
23. Thomson, W. T., Theory of Vibrations with Applications, 2nd Edition, Prentice-Hall, Englewood Cliffs, New Jersey, 1981.
24. Gear, C. William, Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, Englewood Cliffs, New Jersey, 1970.
25. Fresonke, D. A., Hernandez, E., Tesar, D., "Deflection Prediction for Serial Manipulators," Proceedings, 1988 IEEE International Conference on Robotics and Automation, pp 482-487.

